Broadview
www.broadview.com.cn

Java技术大系

看着自己历时近一年的工作，不禁思绪万千……

NetBeans 权威指南

- 激情之作，毫无保留
- Sun公司技术团队审校本书

刘斌 费冬冬 丁璇 编著

光盘包含：
NetBeans 6.0安装文件及
本书所有实例代码



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Broadview®
www.broadview.com.cn
博文视点·IT出版旗舰品牌
技术凝聚实力·专业创新出版



本书受到Sun中国技术社区的认可和推荐！

NetBeans权威指南

第一篇 标准开发环境

第1章 NetBeans的安装和开发环境配置
第2章 通过NetBeans开发Java标准应用程序
第3章 在NetBeans中调试应用程序
第4章 代码重构
第5章 JavaBean开发
第6章 管理和使用服务器
第7章 数据库的管理与开发

第二篇 NetBeans高级应用

第8章 JUnit测试
第9章 性能分析
第10章 在NetBeans中进行版本控制
第11章 基础Web应用程序开发
第12章 高级Web应用程序开发
第13章 在NetBeans中开发Struts程序
第14章 进行可视化的Web开发
第15章 企业级应用程序开发
第16章 基于XML的开发
第17章 UML建模
第18章 Web服务以及SOA程序开发

第三篇 NetBeans手机开发

第19章 创建和运行Java ME应用程序
第20章 Java ME程序的可视化开发
第21章 Java ME开发高级技术

第四篇 NetBeans组件开发

第22章 开发NetBeans插件模块

第五篇 NetBeans开发C++程序

第23章 使用NetBeans开发C/C++程序

源自实践 分享经验

上架建议：程序开发>Java

ISBN 978-7-121-06320-6
9 787121 063206 >

网上订购：www.dearbook.com.cn
第二书店·第一服务

责任编辑：江 立
责任美编：谢丹丹

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书。

定价：75.00元（含光盘1张）

TP312/2858D
2008

Java技术大系

NetBeans 权威指南

刘斌　费冬冬　丁璇　编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

NetBeans 是当前市场上最为流行的、源代码开放且免费的集成开发环境。NetBeans 不仅可以作为 Java 开发工具，还可以使用它开发标准的 Java 程序、Web 应用程序、Java EE 企业级应用程序、Java ME 手机应用程序等，还能够开发 C++程序、Ruby 程序等。

本书主要介绍 NetBeans 6.0 简体中文版集成开发环境的基本功能和使用方法。读者在阅读本书的同时，不仅能够掌握 NetBeans 的使用方法，还能够学习如何进行项目的开发。本书共分为五篇，第一篇介绍标准 Java 应用程序的开发，以及 NetBeans 集成开发环境基本功能。第二篇为 NetBeans 集成开发环境的高级应用篇，本篇结合具体的实例讲解如何进行 JUnit 单元测试、代码性能分析、代码版本控制、开发 Web 应用程序、企业级应用程序、进行 UML 建模，以及开发 SOA 应用程序。第三篇介绍如何使用 NetBeans 进行快速的手机程序开发。第四篇描述的 NetBeans 插件模块和富应用程序的开发方法。第五篇给出了如何在 NetBeans 中进行 C/C++程序的开发方法。

本书内容深入浅出，实例丰富，可帮助读者快速入门，适合希望使用 NetBeans 进行开发的初、中、高级读者，可作为必备的开发工具书使用。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

NetBeans 权威指南 / 刘斌，费冬冬，丁璇编著. —北京：电子工业出版社，2008.5
（Java 技术大系）
ISBN 978-7-121-06320-6

I. N… II. ①刘… ②费… ③丁… III. JAVA 语言－程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2008）第 044065 号

责任编辑：江 立
印 刷：北京天宇星印刷厂
装 订：三河市皇庄路通装订厂
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×1092 1/16 印张：39.5 字数：983 千字
印 次：2008 年 5 月第 1 次印刷
印 数：5000 册 定价：75.00 元（含光盘 1 张）

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

Download at http://www.pin5i.com/



序 Preface

今天本书终于完稿，看着自己历时近一年的工作，不禁思绪万千。在与 Sun 公司的一位朋友聊天时，他提到了 Sun 公司想出版一本关于 NetBeans 的书。我碰巧一直在使用 NetBeans，而且国内当时尚无专门关于 NetBeans 的书。出于好胜之心（也的确低估了其中的工作量），我就答应了下来。

在开始写作后，才发现确实不是一件轻松的活。为了能够写好这本书，从书的结构到内容安排，都经过了精心的策划。在和 Sun 公司的工程师反复商议后，才定下来了大致的框架和结构。在内容上，每章都经过了多次的审阅和修改。这样以来，进度就自然不会快了，再加上俗务缠身，只有晚上才能静下心来写上一些内容。因此导致了本书大约历时一年。说实话，在这一年中，我有好几次打算放弃。要不是答应了 Sun 公司的朋友以及出版社的胡辛征老师，必须履行诺言，而且我也认为这是一次很好的学习过程，否则本书就不能问世了。还好，总算坚持了下来。这里，要非常感谢 Sun 公司的工程师以及出版社老师们的宽容和忍耐，并没有在时间上太催促我。

在写序的时候，又想起了出版社胡辛征老师对我提出的要求，要尽量“活泼”一些，不要墨守成规。工科出身的我，国文的学习只到高中就打住了。以后的十多年中，我把大部分时间花在了“背英文单词→考试→忘记→背英文单词……”的不停循环中。国文的使用，基本上就限于发发邮件而已，而且还是借助于拼音输入法。现在不说提笔忘字吧，也差不了多少了。要像写散文一样写序，可真是让我头痛。不过再困难，工作还得干，毕竟这稿件历时一年多，就差这最后一笔了。于是乎，就出了这篇攒鸡毛凑掸子的序了。还希望大家不要笑话我这三脚猫的国文水平。

逐鹿中原

自从 Java 语言变得越来越热门，成为可以和微软的.NET 分庭抗礼一门主要开发语言后，Java 开发工具的市场也成了诸多公司眼中的大蛋糕。于是乎，群雄纷起，开始逐鹿中原。除了本书中介绍的 NetBeans 以外，还有如下几路英雄。

西楚霸王：JBuilder





在进行角逐的各种势力中，有我们非常熟悉的 Borland（www.borland.com）的 JBuilder。曾记得在大学二年级学习 C 语言时，使用的就是 Borland 的 Turbo C 集成开发环境与 DOS 操作系统。那个时候在学校机房中，使用 1.2MB 的软盘和 Turbo C 开发出了一个中国象棋打谱和管理程序。后来，Windows 逐渐成为主流，Borland 又推出了 Delphi 开发环境。我就从 Turbo C 转移到了 Delphi。从 Delphi 2.0 开始一直到 Delphi 7.0，我一直都是 Delphi 的忠实用户。我也曾为 Delphi 出版过两本书。在寻找 Java 开发工具时，由于对 Borland 的情结，很自然地就会首选 JBuilder 作为开发工具。我想，有我这样因为 Borland 情结而选择 JBuilder 的开发人员不在少数。再加上 JBuilder 推出的时间比较早，这也就使得 JBuilder 很快

Download at http://www.pin5i.com/

就占领了 Java 开发工具的市场，成为这个领域的霸主，长期占有最大的市场份额。

水泊梁山：JDeveloper



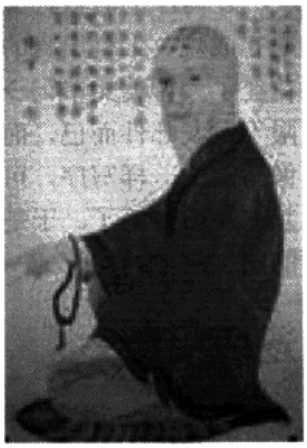
JDeveloper 是数据库领域的老大 Oracle 公司开发的一个 Java 集成开发环境。Oracle 希望将 Java 的部署与开发能力集成到其整个企业级软件产品线之中，JDeveloper 被优化用于同 Oracle 应用和数据库服务器紧密集成，同时它还是一个纯 3GLJava 编程环境。JDeveloper 还是一个开发的开发工具，利用 JDeveloper 建立的 Java 部件可以部署在任何客户、Web 服务器、应用服务器、数据服务器中。

嵩山少林：Visual Cafe for Java



Visual Cafe for Java 是 Symantec（赛门铁克）公司于 1996 年推出的第一个适合于 Java 语言的完整的快速应用开发 RAD（Rapid Application Development）环境。Visual Cafe for Java 类似于 Delphi 开发工具，是 Java 语言可视化快速的集成开发环境，具有图形化用户界面编译器。Visual Cafe for Java 凭借其“七十二绝技”，几乎囊括了所有 Java 开发工具的大奖。

南帝：Visual Age for Java



Visual Age for Java 最早诞生于 IBM 公司在美国卡罗莱纳州北部的 Cary（卡里）开发实验室。由于 IBM 公司的缘故，Visual Age for Java 从一出生就带着雍容华贵的帝王气质，它不仅支持与 WebSphere 的集成开发、EJB 的开发应用等，还可以支持利用 CICS 连接遗传大型机应用。

Visual Age for Java 被包含在了 IBM 著名的 Web 服务器 WebSphere Studio 套件中。凭借其成名绝技六脉神剑（WebSphere）与一阳指（Visual Age for Java），足以令 IBM 纵横 Java 与 Web 开发领域。

神雕大侠：IntelliJ IDEA



IntelliJ IDEA 是 JetBrains 公司（http://www.jetbrains.com/）开发的一个 Java 开发工具。虽然是 Java 开发工具领域的一个较新的成员，但是其功力一点也不逊于其他老前辈。IntelliJ IDEA 支持许多新的特性，包括对 Java EE 的支持、Spring 框架、Hibernate 代码分析、扩展版本控制支持、Rational ClearCase 集成、动态语言支持（例如 Groovy、Ruby、JRuby 等）、插件开发、团队协作等。

瓦岗英雄：WebLogic Workshop



WebLogic Workshop 是 EBA 公司（www.eba.com）推出的一款能够在 BEA WebLogic 服务器上迅速构建、测试和部署面向服务的企业级应用的开发工具。WebLogic Workshop 是一个统一、简化、可扩展

Download at http://www.pin5i.com/

的开发环境，开发人员可以快速、简单地开发出符合客户需要的 Web 服务。同时，WebLogic Workshop 还提供了对 Web 服务的异步调用和回调等先进技术的支持。

令狐冲：Eclipse

Eclipse 最初由 IBM 开发，并在 2001 年 11 月向开源社区开放了全部的源代码，由此变为一个开放源代码的 Java 开发工具。在 Eclipse 开放源代码后（www.eclipse.org），得到了高速的发展，不但功能上得到了大大的提高，而且市场份额也与日俱增，大有问鼎天下之势。

Eclipse 和 Visual Age for Java 都是 IBM 公司开发的，那 IBM 为什么要开发两个 Java 的工具呢？原来，Visual Age for Java 和 WebSphere Studio 集成到一起时会有一些问题和限制，试想把六脉神剑和一阳指拼在一起使用，难免会有问题，不走火入魔已经很不错了。于是，IBM 又“悟”出了又一门神功——独孤九剑。

本来 Eclipse 就身出名门，是天生的练武坯子。又加上一大批武林帮派支持，于是便如日中天，大有一统江湖的趋势。看看 Eclipse 基金会的战略成员，你就知道人家为什么能在那么短的时间内练就傲视群雄的武功了（篇幅有限，只举几个大家比较熟悉的）：

- Borland
- CA
- IBM
- Intel
- Motorola
- Nokia
- Orcale
- Sybase





目前 Eclipse、NetBeans 等争夺武林盟主的战斗仍在激烈地进行着，鹿死谁手言之尚早。

老顽童：Sun Java Studio & JCreator

如果说 NetBeans 是中神通王重阳的话，那么 Sun Java Studio 和 JCreator 就是老顽童。它们不但都是出身一门——全真教（Sun 公司），功力路数也同出一辙（Sun Java Studio 和 JCreator 都是基于 NetBeans 平台的）——都会使用九阴真经。

三分天下

JBuilder 的霸主地位是随着 Eclipse 和 NetBeans 等源代码开放的、免费的 Java 开发工具的兴起而逐渐丧失的。Java 开发工具的市场也逐渐变为 JBuilder、Eclipse 和 NetBeans 三足鼎立。下面的表格中是几个 Java 开发工具的参考报价，可以看出每个开发工具都价格不菲。

产品	价格（美金）
JBuilder 2007 企业版（新用户）	1499
JBuilder 2007 企业版（老用户升级）	750
JBuilder 2007 企业版（支持和维护）	450

Download at http://www.pin5i.com/

Visual Cafe for Java v2.5 专业版	299
Visual Cafe for Java v2.5 数据库开发版	799
IntelliJ IDEA（商业执照）	549
IntelliJ IDEA（个人执照）	249

对于企业和个人开发者而言，除非特殊开发需求，都会自然地倾向于使用免费的开发工具。Eclipse 和 NetBeans 支持的功能逐渐增加和增强，其功能上与商业版的 Java 开发工具已没有大的区别，甚至有些支持的功能还大大超过商业开发工具。在这种情况下，用户的流失是可以预测的必然结果。就像两个高手过招，人家的内力源源不断，而自己的内力消耗一分就少一分，那么随着时间的流逝，结果也就显而易见。

九阴真经 VS 独孤九剑

随着 JBuilder 的兵败乌江，占有市场份额的最大的开发工具变为 Eclipse 和 NetBeans。NetBeans 近几年的快速崛起，不断地挑战着 Eclipse 的市场份额。两强终于要进行华山论剑，决战于紫禁之巅。

我想大家已经读过很多对 Eclipse 和 NetBeans 进行比较的文章，因此这里不再重复地对这些内容进行引述。本文只想给出我个人对于选择 Java 开发工具的观点，以期抛砖引玉。

我经常会被人问到："进行 Java 程序开发，使用什么开发工具好呢？"这使我想起早些国内流行自己买散件组装 PC 的时候，经常有人问我："我们家想装一个电脑，你说什么配置好呢？"这两个看似不相关的问题，其实却有同一个答案，即"你想要用它来干什么"。无论 Java 开发工具也好，PC 机也好，都是一个要使用的工具，因此在选择工具的时候你首先要清楚自己打算用它来干什么。再说得专业点，你得明白你的需求是什么。例如你打算进行图像处理，那么就需要高档的图形加速卡和大屏幕的显示器；如果你打算玩音乐，那么就需要配置高档的声卡；如果你打算进行程序开发，就需用使用快速的 CPU 并增加内存的容量；如果只是进行文字处理，那么就把配置降低，尽量节省 Money。

回到 Java 工具的选择上，我想任何一个工具都有其擅长的地方和不足的地方（相对于其他开发工具）。试想，如果你选择的工具无论其他功能上多么的优秀，但是在你要使用的那部分却不能帮助你，那这个工具对你有多么大的意义呢？因此，我的观点是，选择 Java 开发工具是一个仁者见仁、智者见智的事情，主要可以有如下两个考量点：

① 你要用这个开发工具来干什么？

② 你重视的性能或者功能是哪些？

对于第一个考量点，你要首先弄清楚开发中的几个大的分类。这个分类也有不同的分法，例如按照 NetBeans 下载时候分的开发包可以分为：

- Java SE 桌面程序开发
- Java EE 和 Web 应用程序开发
- SOA
- Java ME 手机应用程序开发
- NetBeans 插件开发
- C/C++开发等

按照技术点来划分，又可以分为：

- 数据库开发

Download at http://www.pin5i.com/

- JSP 开发
- AJAX 开发
- Struts 开发
- Spring 开发
- EJB 开发等

在弄清楚自己的需求后，就需要集中在一个领域中进行比较。至于其他的领域，这个工具即使再优秀也与你无关了。

在某一个领域进行比较的时候，你还得弄清楚你的第二个考量点，即你侧重哪些方面的性能/功能。我想大家都清楚，任何一个工具都不是十全十美的，即使在一个小的领域中，它也是有侧重的。如果它在功能上比较齐全，就可能在速度上受到影响。进行选择的时候，你需要弄清楚你看重哪些，可以忽略哪些。整体上来讲，可以有如下几个常用的主观因素。

- 功能/性能因素：是否有你需要的功能，该工具是否有你在意的 bug 等。
- 兼容性因素：是否和其他可能使用的工具兼容，例如 Web 服务器等。
- 速度因素：工具的运行速度是否符合你的要求。
- 汉化程度因素：如果你对英文不太习惯，那么汉化程序将可能变得非常重要。
- 易用性因素：是否使用起来比较简便，比较顺手。

需要注意的是，以上都是一些主观因素。每个人可能需求的功能不一样，每个人对于易用性的理解也不相同。在选择工具的时候，建议读者多试几个工具，只有通过自己的比较才能得到确切的数据，不要人云亦云。例如就拿易用性因素来说，不管别人介绍使用起来如何简单方便，那些不一定会趁自己的手。

为了让读者更容易理解，我就拿我选择 Java ME 开发工具来举例子，进行具体的说明。下面介绍的内容，仅代表个人观点，如果不符合你的想法，请不要较真。

在最早选择 Java ME 开发工具的时候，由于 Borland 情结，我首先考虑的是 JBuilder。但是不久就放弃了，而选择使用 Eclipse 和 NetBeans。理由很简单，这两个工具在功能上一点也不逊于 JBuilder，而且是免费的。

至于 NetBeans 和 Eclipse，两个工具我都一直在使用。我的另外一本书《Java ME 使用详解——用 Eclipse 进行移动开发》（电子工业出版社），就是使用 Eclipse 进行开发的。但是就我个人而言，如果没有其他特别原因，我会首选使用 NetBeans。我是基于以下原因进行选择的。

1．功能/性能因素

这是我最看重的因素之一，占比重最大。NetBeans 提供了可视化的设计界面，通过拖曳可以完成程序的开发。虽然大多数时候，我还是选择了传统的逐行编写 Code 的方式，这个功能并不经常使用，但是偶尔需要快速创建一个测试程序的时候，我会选择使用这个功能。这个功能被我看做是不可缺少的一个功能。Eclipse 需要安装 EclipseME 插件后才能进行 Java ME 程序开发，并且不能进行可视化的开发，这是 Eclipse 失分的一个地方。

NetBeans 6.0 中提供了一个可视化的游戏设计器，这也是我喜欢的一个功能。相对于 Eclipse 这个功能的缺失，NetBeans 又从这里赚了不少的分。

2．兼容性因素

这是对我来说最重要的一个考虑因素。Eclipse 在这个方面做得就不太好。EclipseME 和摩托罗拉的一些手机模拟器就有兼容性问题，在 EclipseME 中导入摩托罗拉模拟器时会丢失一些

Download at http://www.pin5i.com/

模拟器。但是当使用 NetBeans 进行导入时，所有的模拟器都可以导入。

3．速度因素

如果对比 Eclipse 和具有全部功能的 NetBeans 的启动速度，会发现 Eclipse 比 NetBeans 快一些。在这个因素上，我个人认为 Eclipse 是得分的。但是需要提醒读者的是，这个对比有一定的局限性和主观性，因为这个对比的起始条件是不对等的。即使对于同一个软件，支持 10 个功能的速度，显然要比支持 8 个功能的速度要慢些。笔者曾经试过，下载并安装支持 Java SE 和 Java ME 的安装包，然后再卸载掉一些不使用的模块，速度显然大大提高。不知道有没有人试过让两个工具都只支持 N 个功能，然后看看其启动和运行速度？这确实是我一直想知道的结果。

相对于功能和兼容性因素来说，速度因素我不太重视，因为大多数情况下，我电脑的配置都比较高档一些，这个因素对我进行程序开发几乎没有影响。还有些人说 NetBeans 占有的内存比 Eclipse 多，这个因素我也选择了忽略，因为我机器的内存足够大，这个因素对我几乎没有影响。

客观地说，NetBeans 6.0 比 NetBeans 5.0 慢了一些，因此这里建议 NetBeans 开发社区把速度因素多考虑一些，当然功能多且速度快是最理想的了。如果 NetBeans 变得越来越慢，那势必会影响到很多人来选择 NetBeans。

4．汉化程度因素

Eclipse 的汉化做得有些不如 NetBeans，需要单独下载并安装语言包。从汉化的程度上来说也并不如 NetBeans 彻底。虽然我能够轻松地使用英文版界面工具，并顺利地阅读技术文档，但是我还是偏重于使用中文。就像移居到美国的大多数中国人一样，尽管遍地都是印度、意大利、墨西哥菜，但仍然保留以中餐为首选的习惯。

国内的有些程序员，选择非英文的书不看。（因为国内的计算机书太烂了？）我只是在找不到中文资料的情况下，才去查找英文资料。因为，毕竟从小阅读几乎全部的金庸和古龙先生作品练就的阅读速度，是仅读过几本英文名著（还是靠字典的帮忙）的阅读速度不可比的。

在这个方面，我是给 NetBeans 加分的，也希望它们继续努力！

5．易用性因素

就我个人来讲，我不太喜欢 Eclipse 的项目组织方式，我更倾向于 NetBeans 的项目管理方式。在 NetBeans 中打开一个项目后，只需要按下 F6 键，就可以在模拟器上运行这个项目。而在 Eclipse 中，你还得先打开“运行”对话框，新建或者修改一个配置以后，才能运行这个项目。尽管这个配置过程只需要一次，但是当你经常打开/关闭项目的时候，就会发现这个很烦琐。

在 NetBeans 中，MIDlet 的 JAR 和 JAD 文件都是自动被创建的。在 Eclipse 中要创建 JAR 和 JAD 文件，需要每次在“包资源管理器”窗口中右键单击项目根节点，选择弹出菜单中的 J2ME→Create Package 才能完成。如果你需要不停地修改代码，然后把 JAR 和 JAD 文件安装到手机上进行测试，那么你就会发现这个过程在 Eclipse 会让你感到很厌烦。

如果想清理一个项目，在 NetBeans 中只要在“项目”窗口中右键单击项目根节点，选择“清理”命令即可。在 Eclipse 中则要选择“项目”→“清理”命令，打开“清理”对话框。然后在这个对话框中选择“清理所有项目”还是“清理下面所选的项目”（如果选择这个选项，还要在下面通过复选框选择要被清理的项目）。这个过程也时常让我感到很不舒服。

以上都是我个人在使用过程中发现的对于我适用的易用性因素。

Download at http://www.pin5i.com/

6. 其他因素

其他因素是我认为可以忽略的因素。例如有些文章上说 Eclipse 内核采用的是 OSGi R4 内核框架规范，因此具有高度的可扩展性，这个框架如何先进等。我个人认为这不是我关心的内容，这应该是开发 Java 工具的人来关心的内容。我不管工具的将来如何，现在能不能方便地开发出 Java ME 程序才是我最优先考虑的（黑猫还是白猫，抓住老鼠就是好猫）。

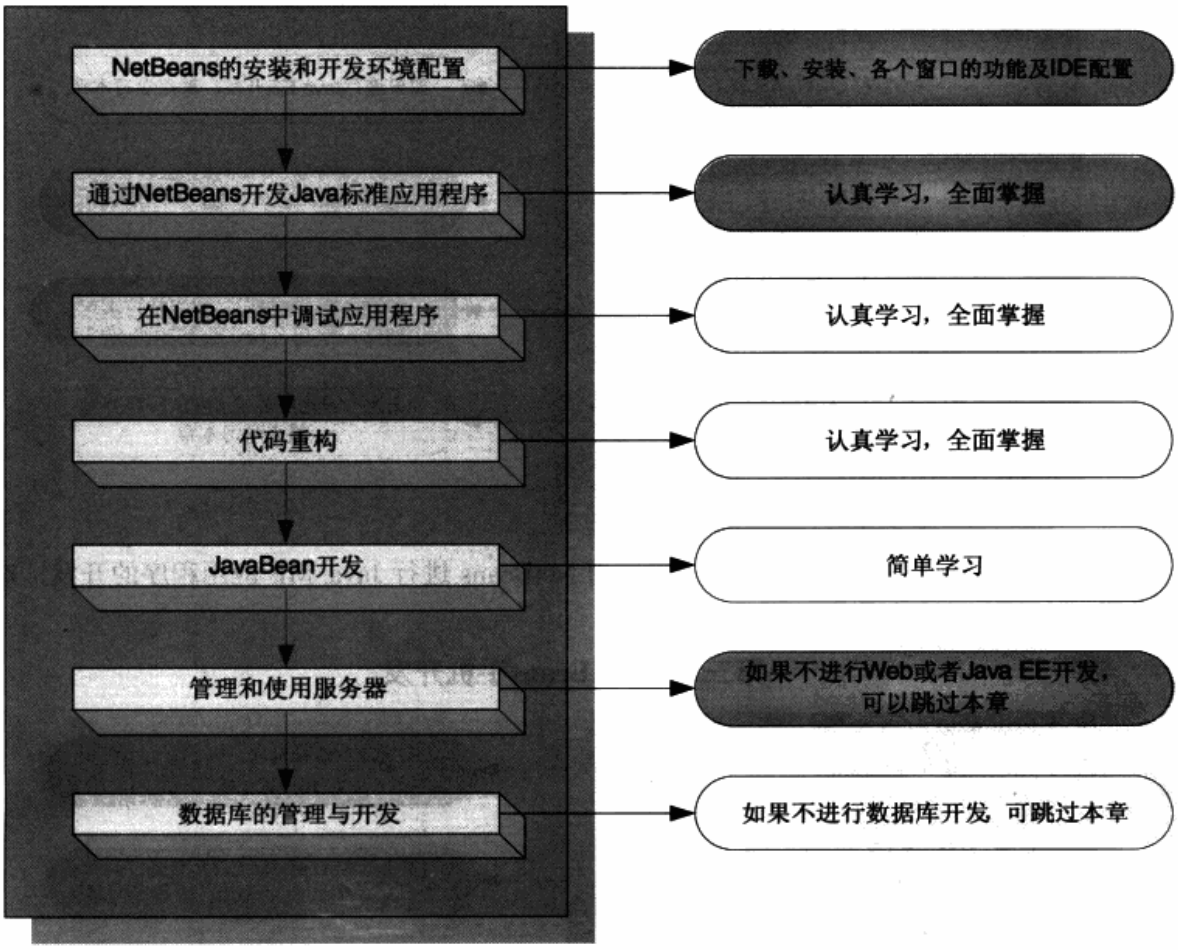
还有些文章中说 Eclipse 因为采用的是 SWT 库，因此界面比 NetBeans 漂亮，这个也是我选择忽略的因素。

沙场秋点兵

本书的学习流程图

本书共分为 5 篇，其中第一篇为标准开发环境，主要讲解 NetBeans 的基本开发环境，在 NetBeans 中开发、调试标准 Java SE 的方法。这部分内容是其他部分内容的基础，因此必须认真掌握。

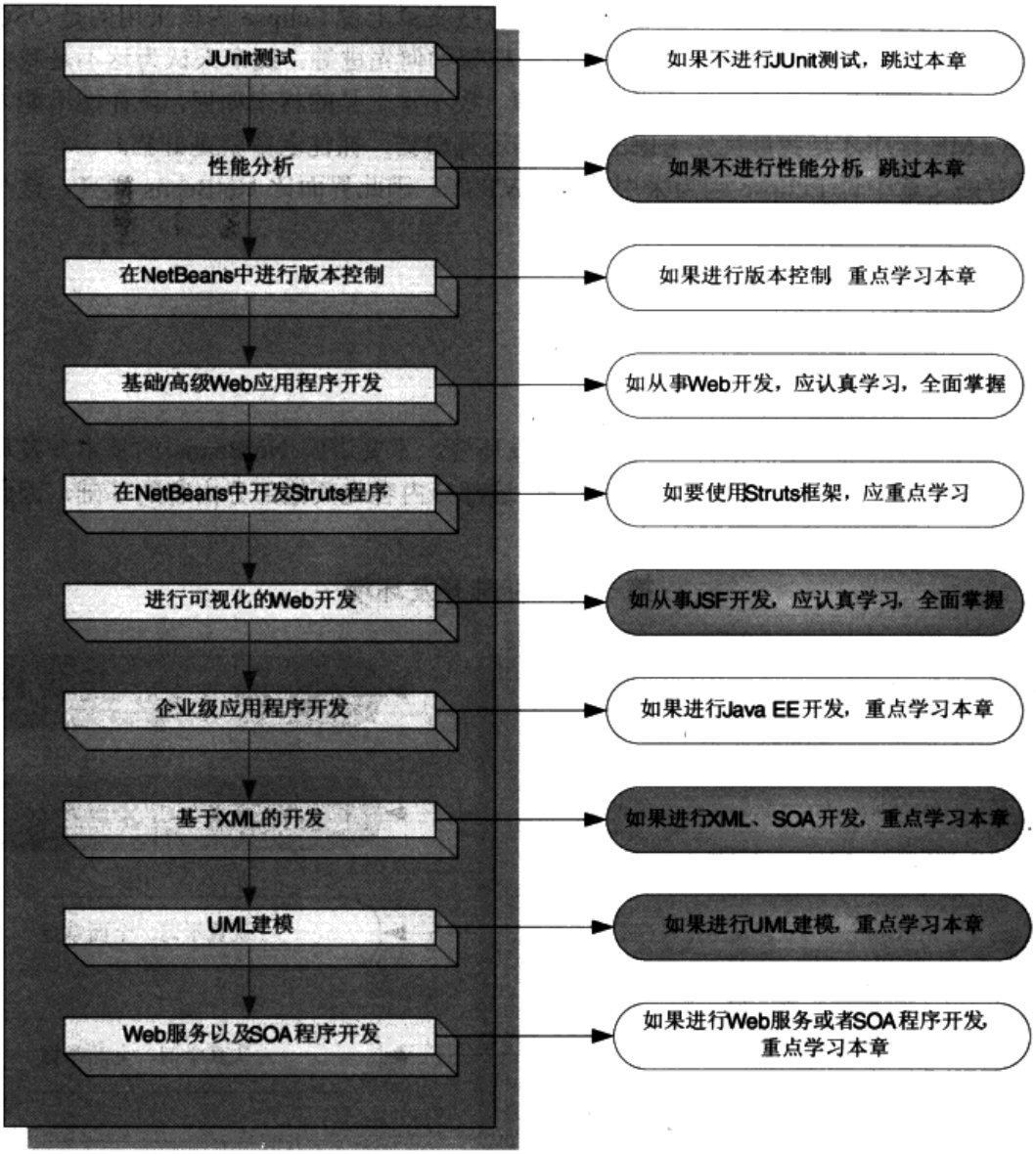
第一篇　标准开发环境



第二篇为 NetBeans 高级应用，这部分主要讲解 Web 开发、Java EE 开发、UML 建模、JUnit 测试等技术，为本书的重点部分。

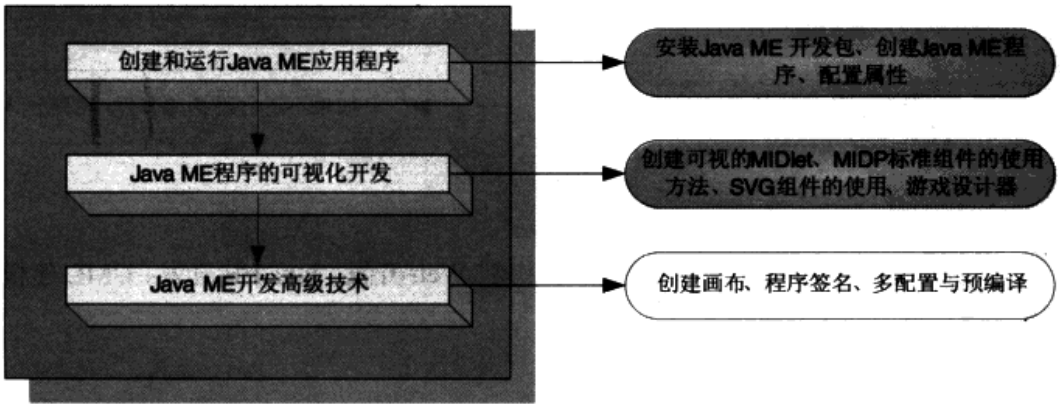
Download at http://www.pin5i.com/

第二篇　NetBeans高级应用



第三篇为手机开发篇，主要讲解如何使用 NetBeans 进行 Java ME 应用程序的开发，对于不进行 Java ME 开发的人员，可以跳过本篇。

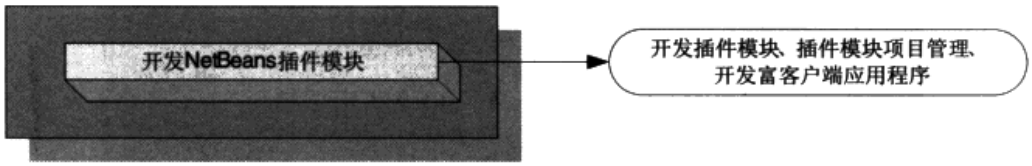
第三篇　NetBeans手机开发



Download at http://www.pin5i.com/

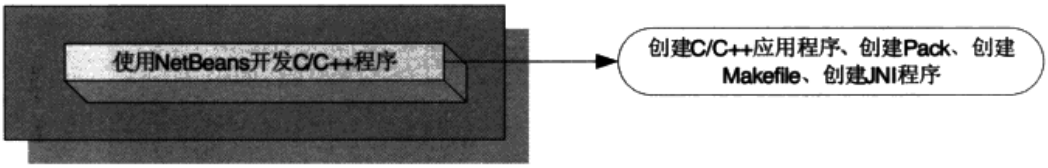
第四篇为 NetBeans 组件开发，这里大概介绍了简单的插件模块开发方法和富客户端程序的开发方法。

第四篇　NetBeans组件开发



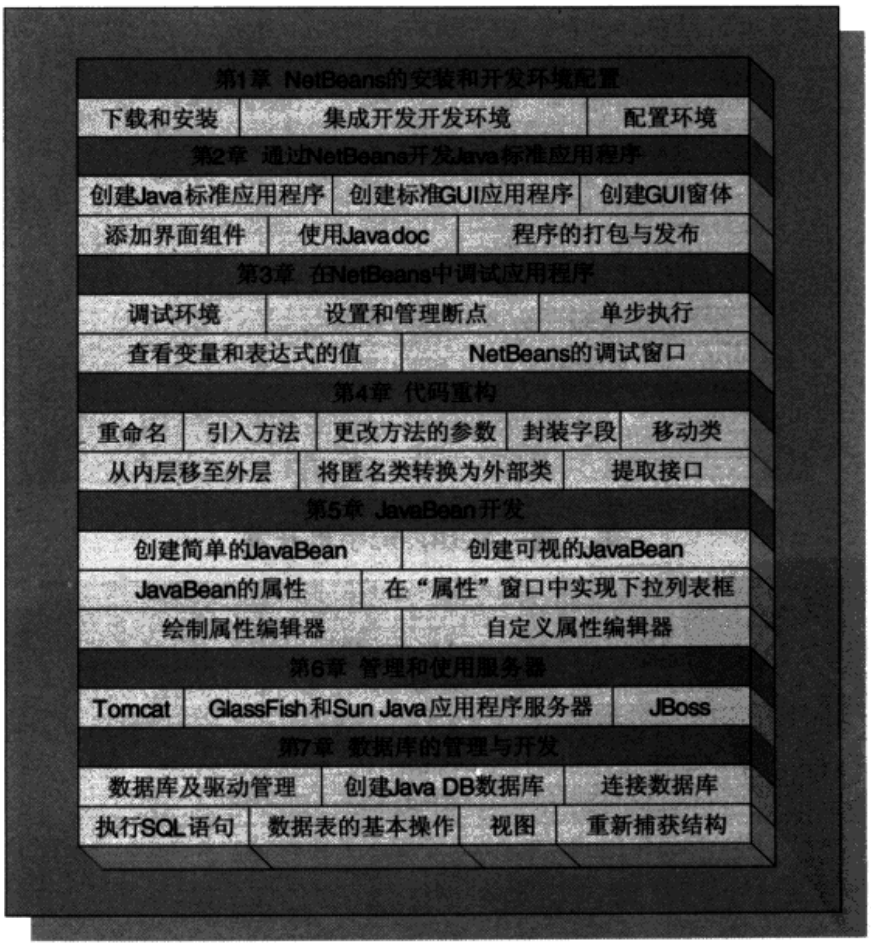
第五篇为 NetBeans 开发 C++程序篇，主要讲解如何在 NetBeans 中开发 C/C++程序。

第五篇　C/C++程序开发



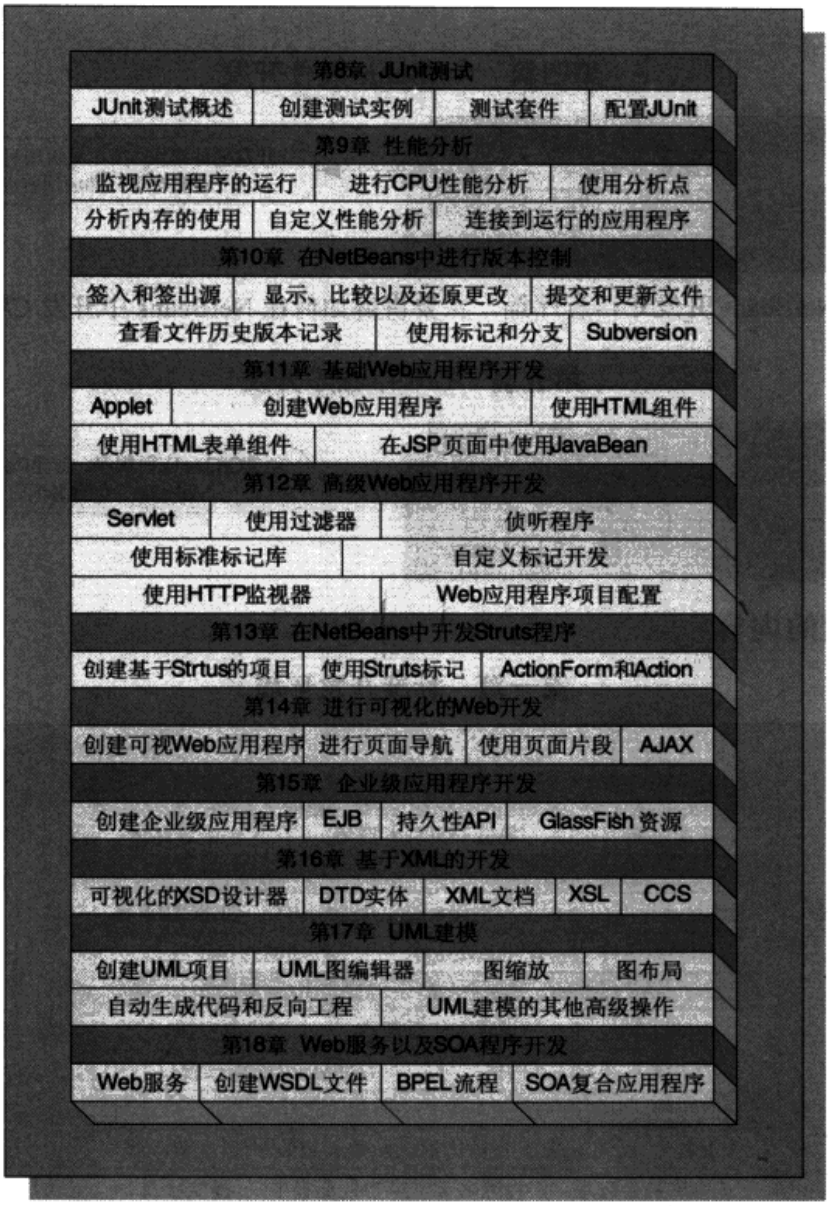
本书包括的内容

第一篇　标准开发环境

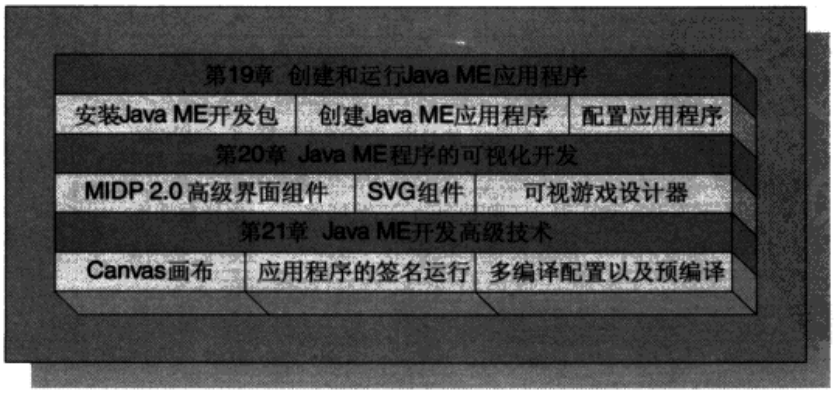


Download at http://www.pin5i.com/

第二篇　NetBeans高级应用

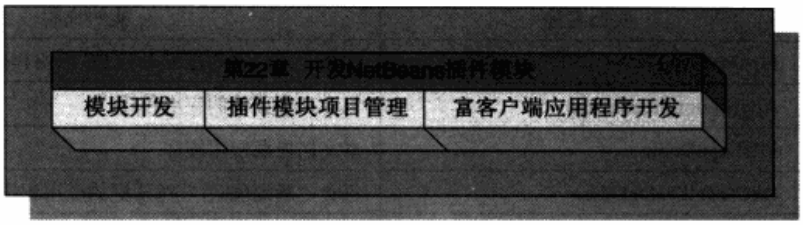


第三篇　NetBeans手机开发

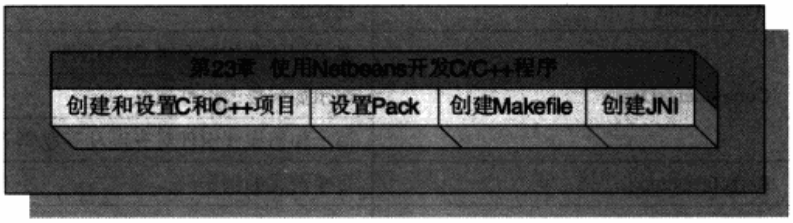


Download at http://www.pin5i.com/

第四篇　NetBeans组件开发



第五篇　NetBeans开发C++程序



和其他书籍相比，本书有何特点

① 本书为 Sun 公司指定 NetBeans 官方用书，书的出版和推广都得到了 Sun 公司的官方支持。书中的内容也得到了 Sun 公司的确认和支持。

② 内容广泛、技术新颖：从内容上来讲，覆盖了 Java 程序开发的标准应用程序开发、Java EE 和 Web 开发、Java ME 开发、模块开发，以及非 Java 技术的 C/C++开发。从技术上来讲，都是一些前沿的技术，例如 Struts 程序开发、JSF 程序开发、AJAX 程序开发、SOA 开发、Web 服务、UML 建模、JUnit 测试，Profiler 性能分析、版本控制等。

③ 知识与实例密切结合：为了能使读者迅速理解书中的每个知识点，在每章介绍技术的同时，穿插着具体的实例。这些实例都是作者精心设计的，力争说明该章节的每个技术点，使读者在实践中理解知识。

④ 例子的纵向联系：本书还注意了例子的纵向联系，即一个例子使用不同的技术来实现，这样读者不但在学习这部分知识的时候可以快速理解，而且还能与其他技术进行纵向比较以加深理解。

⑤ 配带代码光盘，加速学习：本书示例中的所有源代码都可以在随书光盘中找到，根据相应的程序说明，读者可以基于书中的讲解并借助代码进行同步学习。

本书超值光盘和实例分布

1．本书实例源文件

包括本书涉及的所有源文件，读者可以在学习的过程中，直接将其复制到开发环境中，提高学习效率。

操作系统：Windows 2000 及其以上版本

开发环境：JDK 6.0、NetBeans 6.0 简体中文版

数据库：Java DB 数据库

服务器：Tomcat、GlassFish

Download at http://www.pin5i.com/

2. 实例分布表

章	节	实例名称	实例说明
2	2.1.1	FirstJavaApp	创建一个简单的标准 Java SE 应用程序
	2.3.1	ApplicationMenuDemo	包含三个菜单的应用程序
	2.3.2	MDIAppDemo	多文档界面应用程序
	2.3.3	OkCancelDialogDemo	包含“确定/取消”对话框的应用程序
	2.4.1	JDialogDemo	包含对话框窗体的应用程序
	2.4.2	JFrameDemo	包含框架窗体的应用程序
	2.4.3	JPanelDemo	包含的应用程序
	2.5.2	ComponentsDemo	演示如何添加并编辑 GUI 组件
	2.5.3		演示如何修改组件属性
	2.5.4		演示如何设置组件对齐以及等宽/等高
4		RefactorDemo	贯穿全文的例子
5	5.2.1	SimpleBeanDemo	创建一个简单的 JavaBean，以及如何使用 JavaBean
	5.3.1	GuageBean	创建以及使用可视化的 JavaBean 组件
	5.4.3	MyJavaBean	创建带有 Bound 属性的 MyJavaBean 组件
	5.5	MyBeanPropertyEditors	创建简单属属性编辑器
7	7.7	GetDBStructure	从数据库重新捕获结构
8	8.2.2	JUnitTestDemo	进行现有类的测试
9		AnagramGame	进行性能分析的实例程序
11	11.1.1	AppletDemo	创建不带可视化编辑界面的 Applet
	11.2.1	MyWebApplication	创建一个简单的 Web 应用程序
	11.3	HTMLDemo	演示在页面中使用 HTML 组件和表单组件
	11.5	JSP_Login	本节通过一个完成的登录模块实例，详细说明如何在 NetBeans 中开发基于 JSP 的 Web 应用程序。本实例会在后面章节中不断出现
	11.6	JSP_JavaBean_Login	本实例是登录模块的另外一个实现版本，在这个实例中采用 JavaBean 连接后台数据库，即通过 JSP + JavaBean 技术进行开发
12	12.1.2	JSP_JavaBean_Servlet_Login	本实例是登录模块的另外一个实现版本，在这个实例中采用 Servlet 来处理界面提交数据，JavaBeans 连接后台数据库。即通过 JSP + Servlet + JavaBean 技术进行开发，是一个典型的 MVC 结构的应用
	12.2.2	FilterDemo	在前面的几个登录模块的实例中，都只介绍了如何进行登录校验，并没有增加任何的安全措施。事实上，如果用户知道要访问的网页的地址，在不登录的情况下，直接访问这些页面地址都是可以“绕过”登录检验部分的。本实例在第 12.1.2 节的实例基础上，利用过滤器为登录模块增加一个“防火墙”，把没有登录的非法访问进行屏蔽
	12.4.2	JSTL_Demo	演示如何使用库标记

Download at http://www.pin5i.com/

章	节	实例名称	实例说明
	12.4.4	Login_Tag	本实例是登录模块的另外一个实现版本，即 SQL 标记库来访问数据库并判断是否登录成功
	12.5.1	UserTagFile	通过标记文件实现标记开发
	12.5.2	Login_MyTag	本实例是登录模块的另外一个实现版本。首先创建一个用于登录的标记，然后通过这个标记来实现登录模块
13		StrutsDemo	本实例是登录模块的另外一个实现版本，演示如何通过 Struts 框架创建登录模块
14	14.2	VisualSimpleApp	创建简单的 Visual Web 应用程序
	14.3.1	PageNavigation	建立一个简单的可视化的 Web 应用程序，在程序中创建三个 Web 页面，然后设置三个页面之间的导航
	14.3.2	PageNavigation_Dropdown_Button	更复杂一些的页面导航的实现，即通过下拉列表框来选择要打开的页面。通过下拉列表框选择要打开的页面，是很多网站都使用的一种页面导航方式
	14.4	FragmentDemo	演示公司图标的页面片段的实现方法，以及在页面中的引用方法
	14.5	JSF_Login	本实例是登录模块的另外一个实现版本，即通过 JSF 组件实现登录模块
	14.6.2	AjaxDemo	在页面中使用 AJAX 日历组件
15	15.1.1	SimpleEnterpriseAppDemo	演示开发企业级应用程序的一般方法
	15.1.4	DeployableSimpleEADemo	企业应用程序的打包和部署
	15.2.2	CMPBeanDemo	演示如何创建实体 Bean，以及如何创建实体 Bean 的会话 Bean
	15.2.3	MessageBeanDemo	创建一个简单的消息驱动 Bean，打印客户端发送过来的消息
	15.3	PersistentDemo	演示如何创建持久性单元，以及如何使得持久性单元和实体类协同工作
16	16.2~16.6	XMLSampleApp	创建一个样例 XML 架构
	16.7	XMLTest	创建 OASIS XML 目录文件和 XML 解析实体
17	17.2.1	UMLProjectDemo	创建 UML 项目
	17.7.1	UMLDemoProjectSrc	根据 UML 模型自动生成代码
	17.7.4	UMLProject	使用向导进行逆向工程
18	18.1.2	WebServiceDemo	创建一个简单的 Web 服务
	18.1.3	WebClientDemo	在客户端调用 Web 服务
	18.2	MyBpelModule	创建 BPEL 模块项目
	18.5.1	MyCompositeApp	演示如何进行 SOA 复合应用程序的开发
	18.5.2	BPELWebClient	创建 Web 客户端测试 SOA 程序
19	19.3	SimpleMIDlet	一个简单的 Java ME 应用程序
20	20.1	VisualMobile	创建可视的 MIDlet
	20.3	TextBoxDemo	基于文本框（TextBox）的应用程序
	20.4	SplashScreenDemo	使用闪屏（SplashScreen）的应用程序
	20.5	TrickerDemo	使用滚动消息栏（Ticker）的应用程序

Download at http://www.pin5i.com/

章	节	实例名称	实例说明
	20.6	AlertDemo	使用警报（Alert）的应用程序
	20.7	WaitScreenDemo	使用等待屏幕（WaitScreen）的应用程序
	20.8	LoginScreenDemo	使用登录屏幕（LoginScreen）的应用程序
	20.8	LoginScreenServer	使用登录屏幕（LoginScreen）的应用程序服务器端
	20.9	SMSComposerDemo	使用 SMS 编写器（SMSComposer）的应用程序
	20.10	ListDemo	使用列表（List）的应用程序
	20.11	FileBrowserDemo	使用文件浏览器（FileBrowser）的应用程序
	20.12	PIMBrowserDemo	使用 PIM 浏览器（PIMBrowser）的应用程序
	20.13.1	FormDemo	使用文本字段组件（TextField）的应用程序
	20.13.2		使用字符串项组件（StringItem）的应用程序
	20.13.3		使用图像项组件（ImageItem）的应用程序
	20.13.4		使用标尺组件（Gauge）的应用程序
	20.13.5		使用日期字段组件（DateField）的应用程序
	20.13.6		使用选项组组件（ChoiceGroup）的应用程序
	20.13.7		使用 TableItem 组件的应用程序
	20.13.8		使用分隔符组件（Spacer）的应用程序
	20.14.4	SVGDemo	使用 SVG 播放器组件（SVGPlayer）的应用程序
	20.14.6		使用 SVG 菜单组件（SVGMenu）的应用程序
	20.16.1	GameBuilderDemo	NetBeans 游戏生成器的样例游戏
	20.16.2	MyGameBuilderDemo	使用游戏生成器创建的简单游戏
21	21.1	CanvasDemo	创建 Canvas 画布，以及如何绘制画布
	21.2	ListRoots	在讲解应用程序的签名运行时使用的一个预先创建的示例程序
	21.4	MultiConfig	演示如何使用多编译配置以及预编译
22	22.2	MyFirstModule	演示 NetBeans 模块开发方法。在这个实例中创建了一个 NetBeans 菜单命令和工具栏按钮
	22.5	MyModuleSuite	如何开发一个基于 NetBeans 平台的独立的应用程序，即富客户端应用程序开发。该程序有自己的启动闪屏界面
23	23.3	HelloWorld	一个简单的 C/C++应用程序
	23.5.2	TestMakefile	使用 Makefile 向导创建 Makefile
	23.6	HelloWorld_1	创建基于现有代码的 C/C++项目
	23.7.2	Hello	可供 C/C++项目调用的 Java 应用程序
	23.7.3	HelloNative	创建 C/C++动态库项目

适合阅读本书的读者

本书的 5 个部分按照由独立讲解到结合应用的顺序，使读者可以先各自了解每个技术的知识，然后再结合实现程序的开发。本书第一篇为基础篇，其中部分内容必须掌握。其他各篇的学习，读者可以根据自身的技术背景来进行选择。例如希望进行 Web 开发，则要阅读第二篇的内容；进行 Java ME 手机开发则选择阅读第三篇的内容。

书中内容安排由浅入深，非常适合希望使用 NetBeans 工具进行开发的读者。本书也可以作为编程开发的参考工具书使用。

Download at http://www.pin5i.com/

致谢

最后，衷心地感谢 Sun 中文技术社区的蒋清野经理、叶亮以及其他参与本书审阅的 Sun 公司的工程师。他们都非常认真地审阅了本书的内容，并提出了非常具体的修改意见和建议。没有他们作为后盾，这本书是不可能面世的。另外，还要感谢出版社的胡辛征老师，在进度一再落后的情况下，他一直在鼓励我。对他长久以来的支持，这里表示衷心的感谢。这里还要感谢出版社的江立老师，她帮助我发现并改正了很多的错误。

仅以此书献与我所深爱的家人、妻子和我可爱的女儿，他们是我奋斗不息的力量源泉。同时感谢所有关心我、爱护我的朋友们!

刘　斌

2008 年 3 月 6 日于美国硅谷

“终于要交付印刷了，脑海中忽然有很多事涌出来，很想说点什么……

……

谢谢刘斌！谢谢 John！”

本书策划

2008 年 4 月 18 日于办公室灯光下

Download at http://www.pin5i.com/



目 录 Contents

第一篇 标准开发环境

第 1 章 NetBeans 的安装和开发环境配置 2

NetBeans 是 Sun 公司使用 Java 语言开发的一个开源工具，主要包括 IDE（集成开发环境）和 Platform（平台）两个主要部分。其中，集成开发环境是在平台基础上实现的，并且平台本身也开放给开发人员直接使用。本书的大部分内容介绍 NetBeans 集成开发环境的使用方法，只在第 5 篇中介绍 NetBeans 平台的开发使用方法。

利用 NetBeans 集成开发环境可以开发标准的 Java 应用程序、Web 程序、手机程序、C++程序等。本章首先介绍 NetBeans 集成开发环境的安装和基本结构，以及如何更新 NetBeans 等内容。

1.1 下载和安装 NetBeans ····2
1.1.1 NetBeans 6.0 新功能及改进 ····2
1.1.2 NetBeans 对系统的最低要求 ····5
1.1.3 下载 NetBeans 6.0 ····6
1.1.4 安装 NetBeans 6.0 全功能包 ····7
1.2 NetBeans 集成开发环境 ····9
1.3 NetBeans 菜单和工具栏 ····10
1.4 NetBeans 窗口 ····12
1.4.1 “项目”窗口 ····12
1.4.2 “文件”窗口 ····13
1.4.3 “服务”窗口 ····14
1.4.4 “输出”窗口 ····14
1.4.5 “导航”窗口 ····15
1.4.6 “组件面板”窗口 ····16
1.4.7 “属性”窗口 ····17
1.4.8 “检查器”窗口 ····17
1.4.9 “任务列表”窗口 ····18
1.4.10 “收藏夹”窗口 ····19
1.5 源代码编辑器 ····20
1.5.1 快速编写代码 ····20
1.5.2 代码模板 ····21
1.5.3 格式化 Java 代码 ····21
1.5.4 代码跳转 ····22
1.6 更新 NetBeans ····23
1.7 配置 NetBeans 开发环境 ····23
1.7.1 常规配置 ····24
1.7.2 编辑器的常规选项 ····24

Download at http://www.pin5i.com/

1.7.3 编辑器的缩排选项……25
1.7.4 定制代码模板……27
1.7.5 字体和颜色……27
1.8 小结……28

第 2 章 通过 NetBeans 开发 Java 标准应用程序 29

通过 NetBeans 集成开发环境可以快速地创建 Java 标准应用程序，利用 NetBeans 提供的可视化编程界面可以为应用程序添加/编辑界面组件、编译运行项目并最后发布项目。本章将讨论如何在 NetBeans 集成开发环境中完成这些基本的任务，快速开发 Java 标准应用程序。本章的讲解将以一个实际项目的开发过程为基础，一步一步地介绍 Java 标准项目在 NetBeans 集成开发环境中的开发实现。

2.1 创建 Java 标准应用程序……29
2.1.1 创建 Java 标准应用程序项目……29
2.1.2 打开和关闭项目……32
2.1.3 导入项目……32
2.1.4 重命名项目……36
2.1.5 项目的复制、移动和删除……37
2.1.6 创建 Java 主类……38
2.1.7 创建 Java 包……39
2.2 项目运行与项目配置……40
2.2.1 项目运行……40
2.2.2 项目配置……41
2.3 创建标准 GUI 应用程序……44
2.3.1 包含三个菜单的应用程序……44
2.3.2 多文档界面应用程序……45
2.3.3 “确定”/“取消”对话框……46
2.4 创建 GUI 窗体……48
2.4.1 对话框窗体……49
2.4.2 框架窗体……49
2.4.3 面板窗体……50
2.5 添加界面组件……51
2.5.1 GUI Builder……51
2.5.2 添加/编辑组件……52
2.5.3 修改组件属性……54
2.5.4 设置组件对齐以及等宽/等高……57
2.5.5 布局管理器和 GridBagLayout 布局……58
2.6 使用 Javadoc……62
2.6.1 设置 Javadoc 生成参数……62
2.6.2 生成 Javadoc……63
2.7 程序的打包与发布……63
2.7.1 设置包装选项……63

Download at http://www.pin5i.com/

2.7.2 项目的生成……64
2.8 小结……64

第 3 章 在 NetBeans 中调试应用程序 65

在开发程序的时候，可能会出现各种错误，虽然编译器会帮助开发人员找到一些语法错误，但是对于一些逻辑错误或者算法上的错误，就只能利用 NetBeans 的调试技术进行分析和解决了。本章将讨论 NetBeans 如何调试应用程序、进行单步跟踪、查看本地变量等内容。

3.1 在 NetBeans 中调试应用程序……65
3.2 设置和管理断点……66
3.2.1 断点类型……66
3.2.2 添加行断点……67
3.2.3 添加类断点……67
3.2.4 添加方法断点……68
3.2.5 添加异常断点……69
3.2.6 添加线程断点……69
3.2.7 添加字段断点……70
3.2.8 断点的管理……70
3.3 单步执行……71
3.4 查看变量和表达式的值……72
3.4.1 在代码编辑器中直接查看……72
3.4.2 使用“监视”窗口查看……73
3.4.3 使用“局部变量”窗口查看……73
3.5 计算表达式……74
3.6 在调试模式修改代码……75
3.7 NetBeans 的调试窗口……75
3.7.1 “装入的类”窗口……75
3.7.2 “调用栈”窗口……76
3.7.3 “线程”窗口……76
3.7.4 “会话”窗口……77
3.7.5 “源”窗口……77
3.8 小结……78

第 4 章 代码重构 79

在程序开发中经常需要修改代码，例如修改变量名。如果一个变量在不同文件的不同地方被使用，那么每个地方都需要修改。进行手工修改很容易发生错误，NetBeans 提供了代码重构的功能，可以帮助开发人员迅速地修改代码而不发生错误。

4.1 重命名……79
4.1.1 重命名项目……79
4.1.2 重命名其他元素……80
4.2 引入方法……82
4.3 更改方法的参数……83

Download at http://www.pin5i.com/

4.4 封装字段……84
4.5 移动类……86
4.5.1 使用复制/粘贴法移动类……86
4.5.2 重构的移动类命令……86
4.6 从内层移至外层……87
4.7 将匿名类转换为外部类……88
4.8 提取接口……89
4.9 小结……91

第 5 章 JavaBean 开发 92

代码复用一直是开发人员追求的一个目标，组件技术是实现代码复用的一个很好途径。目前流行的组件技术有 COM、DCOM、CORBA 等。JavaBean 是利用 Java 实现的组件技术，具有 Java 语言所有的特点，如易用性以及平台无关性。JavaBean 的设计目标就是一次性编写、任何地方执行、任何地方重用。

本章主要介绍如何在 NetBeans 中开发和利用 JavaBean 组件。首先介绍 JavaBean 的基本知识，然后通过实例介绍如何开发一个简单的 JavaBean，以及如何使用这个 JavaBean 组件。接着，逐步深入给出可视化 JavaBean 的开发方法，如何为 JavaBean 添加属性等内容。在学习本章前，建议读者先认真学习一下 JavaBean 开发的相关知识。

5.1 JavaBean 概述……92
5.2 如何创建一个简单的 JavaBean……93
5.2.1 创建一个简单的 JavaBean……93
5.2.2 如何使用 JavaBean……95
5.3 创建和使用可视化的 JavaBean 组件……96
5.3.1 创建可视化的 JavaBean 组件……96
5.3.2 使用可视化的 JavaBean 组件……100
5.4 JavaBean 的属性……101
5.4.1 创建简单属性……101
5.4.2 Indexed 属性……102
5.4.3 Bound 属性……103
5.5 创建简单属性编辑器……103
5.6 在“属性”窗口中实现下拉列表框……105
5.6.1 创建 MyShape 组件……105
5.6.2 创建下拉列表框形式的属性编辑器……106
5.6.3 将属性和属性编辑器关联……108
5.7 绘制属性编辑器……109
5.8 自定义属性编辑器……110
5.9 小结……113

第 6 章 管理和使用服务器 114

服务器是供应用程序运行的计算机。通过安装服务器软件并将计算机连接至互联网，可以使任何计算机成为服务器。在 NetBeans 集成开发环境中支持多个服务器，例如 Java EE 服务器、

Download at http://www.pin5i.com/

Web 服务器等。本章将讨论如何在 NetBeans 中注册和管理一些常用的服务器，例如 GlassFish、Sun Java Application Server、Tomcat 等。

6.1 Tomcat 服务器……114
6.1.1 启动和停止服务器……114
6.1.2 编辑 Tomcat 配置文件……115
6.1.3 定制 Tomcat 服务器……115
6.1.4 Tomcat 的管理控制台……118
6.1.5 注册外部的 Tomcat 服务器……119
6.2 GlassFish 和 Sun Java 应用程序服务器……120
6.2.1 安装 Sun Java 应用程序服务器……121
6.2.2 注册 Sun Java 应用程序服务器……123
6.2.3 修改 Sun Java Application Server 的 JVM 属性……125
6.2.4 管理 GlassFish & Sun Java Application Server……126
6.3 JBoss 应用程序服务器……128
6.3.1 JBoss 服务器的注册……128
6.3.2 JBoss 服务器的管理控制台……129
6.4 小结……129

第 7 章 数据库的管理与开发 130

NetBeans 集成开发环境提供了对数据库开发的支持，用户可以执行 SQL 语句或者可视化地对数据库进行操作。NetBeans 对目前大多数的主流数据库都提供了支持，例如 SQL Sever 数据库、Access 数据库、MySQL 数据库、Java DB 数据库等。

本章按照使用的顺序，介绍了数据库驱动程序的配置、数据库的连接、数据表的操作以及如何执行 SQL 语句。

7.1 数据库及驱动管理……130
7.2 创建 Java DB 数据库……131
7.3 连接数据库……132
7.3.1 连接 Java DB 数据库……132
7.3.2 Pointbase 数据库的注册和连接……134
7.3.3 连接 MySQL 数据库……134
7.4 在 NetBeans 中执行 SQL 语句……136
7.5 数据表的基本操作……137
7.5.1 创建数据表……137
7.5.2 浏览表的结构……138
7.5.3 添加/删除列……138
7.5.4 抓取结构以及重新创建表……140
7.6 视图操作……141
7.7 从数据库重新捕获结构……142
7.8 小结……144

Download at http://www.pin5i.com/

第二篇　NetBeans 高级应用

第 8 章　JUnit 测试　146

众所周知，测试在整个软件生命周期中占有很重要的地位。单元测试又称为模块测试，是对一个模块或者一个单元进行的测试，因此称为单元测试。NetBeans 提供了对程序进行单元测试的支持，用户可以通过 JUnit 快速完成程序单元的测试。本章将首先介绍一些 JUnit 的基础知识，然后介绍如何进行单元测试。

8.1　JUnit 测试概述 ……146
8.2　创建测试实例 ……147
8.2.1　创建空测试 ……147
8.2.2　进行现有类的测试 ……148
8.3　测试套件 ……151
8.4　配置 JUnit ……152
8.5　小结 ……153

第 9 章　性能分析　154

在开发应用程序时，有时需要知道代码的执行效率如何以及内存的占用情况，以便于对代码进行优化、提高应用程序的性能。因此，在对代码进行优化前，对代码进行性能分析就显示十分必要。NetBeans 提供了一个优秀的代码分析工具 Profiler，本章将详细地介绍如何利用这个工具进行性能分析、监视应用程序的运行等。

9.1　Profiler 概述 ……154
9.1.1　安装 Profiler 工具 ……154
9.1.2　校准 Profiler ……155
9.1.3　Profiler 控制面板 ……155
9.2　监视应用程序的运行 ……157
9.2.1　启动程序监视 ……157
9.2.2　VM 遥测概览 ……158
9.2.3　“线程”窗口 ……159
9.2.4　查看监控状态和基本监控数据 ……160
9.3　进行 CPU 性能分析 ……161
9.3.1　启动性能分析 ……161
9.3.2　查看快照 ……162
9.3.3　查看性能分析结果 ……166
9.3.4　查看性能分析状态和基本数据 ……166
9.3.5　指定根方法 ……166
9.4　使用分析点 ……167
9.4.1　创建重置结果类型分析点 ……168
9.4.2　创建秒表类型分析点 ……169
9.4.3　创建生成快照类型分析点 ……169
9.4.4　创建定时生成快照类型分析点 ……170

Download at http://www.pin5i.com/

9.4.5 创建触发生成快照类型分析点 ····· 170
9.4.6 使用分析点 ····· 171
9.5 分析内存的使用 ····· 171
9.5.1 启动内存分析 ····· 172
9.5.2 内存快照 ····· 172
9.6 自定义性能分析 ····· 174
9.6.1 创建定制的性能分析配置 ····· 174
9.6.2 自定义监视分析参数 ····· 175
9.6.3 自定义 CPU 分析参数 ····· 176
9.6.4 自定义内存分析参数 ····· 177
9.7 将 Profiler 连接到运行的应用程序 ····· 178
9.7.1 使用连接向导进行直接连接 ····· 178
9.7.2 进行动态连接 ····· 181
9.8 配置 Profiler 选项 ····· 181
9.9 小结 ····· 183

第 10 章 在 NetBeans 中进行版本控制 184

CVS（Concurrent Version System）和 Subversion 都是非常优秀的版本控制系统，它可以在软件开发过程中记录文件的变更细节以及存储软件开发过程中的各种版本。版本控制系统允许多名开发人员同时编辑同一文件而不会丢失数据。每名开发人员都拥有该文件的副本，最后将所有的修改合并到一起成为最终的结果。CVS 和 Subversion 都基于 C/S，可以运行在 UNIX、Linux 以及 Windows 平台之上。CVS（Subversion）的服务器端存有一个源代码库，源代码库中保存众多不同的项目。客户端通过和服务器端建立连接，将源代码下载到本地计算机上，开发人员就可以在本地计算机上进行添加、修改、删除等操作，最后提交到服务器。

NetBeans 内置对 CVS 和 Subversion 的支持，NetBeans 可以作为 CVS 系统的客户端从而对开发进行版本控制。本章以 CVS 为主讲解如何在 NetBeans 集成开发环境中进行版本控制，在本章的最后部分介绍如何在 NetBeans 中使用 Subversion 以及其他的版本控制系统。

10.1 CVS 基本原理 ····· 184
10.2 搭建 CVS 服务器 ····· 185
10.3 签入和签出源 ····· 186
10.3.1 签入 ····· 186
10.3.2 签出 ····· 187
10.3.3 通过 CVS 浏览 NetBeans 源代码 ····· 188
10.3.4 查看文件状态信息 ····· 188
10.4 显示、比较以及还原更改 ····· 189
10.4.1 显示更改 ····· 189
10.4.2 比较更改 ····· 190
10.4.3 还原更改 ····· 191
10.5 提交和更新文件 ····· 191
10.5.1 提交文件 ····· 191
10.5.2 更新文件 ····· 192
10.5.3 解决冲突 ····· 192

Download at http://www.pin5i.com/

10.6 查看文件历史版本记录……193
10.7 使用标记和分支……194
10.7.1 使用标记……194
10.7.2 使用分支……195
10.8 使用 Subversion……196
10.8.1 Subversion 简介……196
10.8.2 下载并安装 Subversion 客户端……197
10.8.3 在 NetBeans IDE 中设置 Subversion……198
10.9 安装 Mercurial……199
10.10 小结……200

第 11 章 基础 Web 应用程序开发 201

Web 应用程序是为互联网编写的应用程序，很多语言都支持 Web 程序的开发，例如 Perl、ASP.NET、Java 等。NetBeans 集成开发环境提供了对 Web 程序开发的良好支持，可以创建空 Web 应用程序项目、快速编写 HTML 页面、建立 JSP 文件、开发 Servlet、使用 JavaServer Faces 和 Struts 框架、开发 Applet 小程序、创建和使用标记等。

本章首先介绍如何开发一个可以在网页上运行的小程序，然后介绍如何开发一个包含 HTML 页面和 JSP 页面的 Web 应用程序。

在学习本章时，假定读者已经具有了编写 JSP 程序的能力，这里着重介绍如何在 NetBeans 集成开发环境中使用这些技术进行开发。对于一些初学者，建议首先学习必要的知识。另外，为了便于初学者学习，对于讲到的一些技术，书中都进行了必要的解释，这样可以保证读者能够顺利完成本章的学习。

11.1 开发 Applet……201
11.1.1 创建小程序……201
11.1.2 运行小程序……207
11.1.3 在 JDK 1.5 中运行小程序……208
11.2 创建一个简单的 Web 应用程序……210
11.2.1 创建 Web 应用程序项目……211
11.2.2 运行 Web 应用程序……213
11.2.3 调试 Web 应用程序……214
11.2.4 生成 Web 应用程序……214
11.2.5 部署 Web 应用程序……215
11.3 在页面中使用 HTML 组件……216
11.3.1 创建 HTML 和 XHTML 页面……216
11.3.2 创建 JSP 页面……217
11.3.3 HTML 组件面板……218
11.3.4 使用 HTML 表……218
11.3.5 使用 HTML 有序列表组件……220
11.3.6 使用 HTML 无序列表组件……222
11.3.7 使用 HTML 图像组件……224
11.3.8 使用 HTML 链接组件……224
11.3.9 使用 HTML 元数据组件……225

Download at http://www.pin5i.com/

11.4 在页面中使用 HTML 表单组件 ……227
11.4.1 使用 HTML 表单 ……227
11.4.2 使用 HTML 按钮 ……227
11.4.3 使用 HTML 文本输入 ……228
11.4.4 使用 HTML 多行输入 ……230
11.4.5 使用 HTML 下拉列表 ……231
11.4.6 使用 HTML 复选框 ……233
11.4.7 使用 HTML 单选按钮 ……234
11.4.8 使用 HTML 选择文件 ……235
11.5 基于 JSP 的开发实例——登录模块的实现 ……236
11.5.1 登录模块实现流程 ……237
11.5.2 设计并创建数据库 ……237
11.5.3 创建 logincheck.jsp 文件 ……238
11.5.4 创建 index.jsp 文件 ……240
11.5.5 创建 welcome.jsp 文件 ……242
11.5.6 创建 error.jsp 文件 ……243
11.5.7 运行项目 ……244
11.6 在 JSP 页面中使用 JavaBean ……245
11.6.1 在 JSP 页面中声明 JavaBean ……245
11.6.2 读取 Bean 属性 ……246
11.6.3 设置 Bean 属性 ……246
11.6.4 基于 JSP + JavaBean 的开发实例——登录模块的实现 ……247
11.7 小结 ……248

第 12 章 高级 Web 应用程序开发 249

上一章介绍了如何使用 Applet 小程序以及 JSP 开发 Web 应用程序。本章将介绍如何开发 Servlet、使用标记等内容，以及如何进行网络监视、项目的打包部署、项目配置等。

在学习本章时，假定读者已经具有了编写 Servlet 等技术的能力，这里主要着重介绍如何在 NetBeans 集成开发环境中使用这些技术进行开发。对于一些初学者，建议首先学习必要的知识。另外，为了便于初学者学习，对于讲到的一些技术，书中都作了必要的解释，这样可以保证读者能够顺利地完成本章的学习。

12.1 开发 Servlet ……249
12.1.1 在 NetBeans 中创建 Servlet ……249
12.1.2 Servlet 使用实例——登录模块 ……252
12.2 使用过滤器 ……253
12.2.1 NetBeans 中创建过滤器的方法 ……253
12.2.2 过滤器使用实例——为登录模块提供安全保护 ……258
12.3 创建侦听程序 ……261
12.4 使用标准标记库 ……261
12.4.1 JTSL 概述以及 NetBeans 对标准标记库的支持 ……262
12.4.2 使用核心库标记 ……262
12.4.3 使用数据库处理标记 ……266

Download at http://www.pin5i.com/

12.4.4 JSTL 开发实例——使用 JSTL 实现登录模块 ····270
12.5 自定义标记开发 ····271
12.5.1 通过标记文件实现标记开发 ····271
12.5.2 通过标记处理程序实现标记开发 ····274
12.6 使用 HTTP 监视器 ····278
12.6.1 打开 HTTP 监视器 ····278
12.6.2 分析会话数据 ····278
12.7 Web 应用程序项目配置 ····279
12.7.1 配置项目源和 Web 文件夹 ····279
12.7.2 为项目添加新框架 ····280
12.7.3 指定项目的类路径 ····280
12.7.4 设置编译选项 ····281
12.7.5 设置打包选项 ····282
12.7.6 进行运行/部署配置 ····282
12.8 配置 context.xml 描述符 ····283
12.9 配置 web.xml 描述符 ····283
12.9.1 常规 ····283
12.9.2 上下文参数 ····284
12.9.3 应用程序侦听程序 ····284
12.9.4 Servlet ····285
12.9.5 过滤器 ····286
12.9.6 欢迎文件 ····287
12.9.7 错误页面 ····288
12.9.8 JSP 属性组 ····288
12.9.9 环境条目 ····289
12.9.10 资源引用 ····290
12.9.11 资源环境引用 ····291
12.9.12 EJB 引用 ····291
12.9.13 消息目标引用 ····291
12.9.14 登录配置 ····293
12.9.15 安全角色 ····293
12.9.16 安全约束 ····294
12.10 配置 sun-web.xml 描述符 ····294
12.10.1 常规 ····295
12.10.2 消息 ····295
12.11 小结 ····296

第 13 章 在 NetBeans 中开发 Struts 程序 297

Struts 项目是由 Apache Software Foundation 发起的开源项目，它在 Java 服务器端实现了 MVC（Model-View-Controller）设计模式。Struts 为使用 MVC 体系编写应用提供了一个架构，因而目前非常流行。

Download at http://www.pin5i.com/

13.1 Struts 概述……297
13.1.1 MVC 设计模式……297
13.1.2 Web 应用程序的 MVC 实现模式……298
13.1.3 Struts 框架结构及原理……299
13.2 创建基于 Struts 的项目……300
13.3 使用 Struts 标记创建 JSP 文件……300
13.4 创建 ActionForm Bean……302
13.5 创建 Action 操作……304
13.6 小结……308

第 14 章 进行可视化的 Web 开发 309

在前面章节中介绍了一般的 Web 应用程序的开发，以及 Struts 框架的使用方法。本章将介绍目前比较流行的一种可视化的开发技术，即使用 JSF（JavaServer Faces）组件进行可视化的 Web 开发。

JSF 是业界流行的一种开发 Web 应用程序的框架，利用该框架可以像设计 Windows 应用程序一样，在 Web 页面中添加组件并处理用户事件。说得形象一些，利用 JSF 技术，用户可以像开发 ASP.NET 应用程序一样进行基于 Java 的 Web 应用程序开发。

14.1 可视化的 Web 开发概述……309
14.1.1 JavaServer Faces 开发概述……309
14.1.2 安装 Visual Web 开发包……310
14.2 创建简单的 Visual Web 应用程序……310
14.2.1 创建可视化的 Web 应用程序项目……311
14.2.2 可视化的 Web 应用程序开发环境……311
14.2.3 进行可视化的页面编辑……312
14.3 进行页面导航……314
14.3.1 简单的页面导航……314
14.3.2 通过下拉列表框进行页面导航……316
14.4 使用页面片段……317
14.5 使用 JSF 实现登录模块……318
14.5.1 实现登录失败的界面……318
14.5.2 实现登录成功的界面……319
14.5.3 登录界面以及身份验证……320
14.6 进行 AJAX 开发……321
14.6.1 下载并导入 AJAX 及其组件……321
14.6.2 AJAX 开发实例：在页面中使用日历……323
14.7 小结……324

第 15 章 企业级应用程序开发 325

在前面的章节中介绍了使用 NetBeans 开发桌面应用程序、Web 应用程序、可视化的 Web 应用程序等。从本章开始将介绍企业级应用程序的开发，及其相关技术。本章将首先介绍一个简单的企业级应用程序的开发方法，以便于读者掌握程序开发的大致步骤。然后在此基础上进一步

Download at http://www.pin5i.com/

扩展，介绍 EJB 模块和持久性 API 的开发方法。最后，简单给出了几种 Sun 资源的创建方法。

15.1 创建一个简单的企业级应用程序……325
15.1.1 创建企业应用程序项目……325
15.1.2 实现 EJB 3.0 Bean……326
15.1.3 在 Web 应用程序中调用 EJB……329
15.1.4 企业应用程序的打包和部署……332
15.2 EJB 模块开发……333
15.2.1 Enterprise Bean 概述……333
15.2.2 开发实体 Bean……334
15.2.3 开发驱动消息 Bean……340
15.2.4 自动生成发送 JMS 消息的代码……343
15.3 使用持久性 API……344
15.3.1 创建持久性单元……344
15.3.2 创建实体 Bean……346
15.4 创建 GlassFish 资源……347
15.4.1 JDBC 连接池资源……347
15.4.2 JDBC 资源……349
15.4.3 JMS 资源……350
15.4.4 JavaMail 会话资源……352
15.5 小结……353

第 16 章 基于 XML 的开发 354

XML 是业界进行数据交换和传输的一种标准协议。NetBeans 的项目配置和编译文件，例如 build.xml、build.xml、build-impl.xml 等都是 XML 文件。NetBeans 提供了对 XML 开发的支持，通过可视化的 XML 架构设计器，可以快速地创建 XML 架构。

本章的重点是如何通过可视化的 XML 架构设计器编写 XML 架构，以及如何开发和编辑 XML 文档。

16.1 NetBeans 对 XML 开发的支持……354
16.2 创建一个样例 XML 架构……355
16.3 使用可视化的 XSD 设计器……356
16.3.1 “构架”标签页……356
16.3.2 “设计”标签页……356
16.3.3 “源”标签页……358
16.3.4 进行查找……358
16.3.5 添加 XML 架构组件……359
16.3.6 查找 XML 架构组件的使用实例……360
16.4 创建和编辑 DTD 实体……361
16.4.1 通过向导创建 DTD 实体……361
16.4.2 检查 DTD 文档……362
16.4.3 生成 CSS 和文档……362
16.5 创建和编辑 XML 文档……363

Download at http://www.pin5i.com/

16.5.1 通过向导创建 XML 文档……363
16.5.2 查看 XML 文档……365
16.5.3 检查 XML 文档……365
16.5.4 验证 XML 文档……365
16.5.5 根据 XML 文档生成 DTD……366
16.6 创建可扩展样式表……367
16.6.1 通过向导创建 XSL 式样表……367
16.6.2 转换 XSL 式样表……369
16.7 创建 OASIS XML 目录文件和 XML 解析实体……370
16.8 创建层叠样式表……370
16.8.1 层叠样式表编辑器的“字体”面板……370
16.8.2 层叠样式表编辑器的“背景”面板……372
16.8.3 层叠样式表编辑器的“文本块”面板……373
16.8.4 层叠样式表编辑器的“边框”面板……374
16.8.5 层叠样式表编辑器的“边距”面板……375
16.8.6 层叠样式表编辑器的“位置”面板……376
16.9 小结……377

第 17 章 UML 建模 378

统一建模语言 UML（Unified Modeling Language）是为团队开发提供的标准通用的设计语言，也就是像建筑行业使用的建筑图纸一样。目前，UML 已经获得了业界的广泛认同，越来越多的系统都使用 UML 来进行设计和开发。

NetBeans 提供了对 UML 的全面支持，利用 NetBeans UML 建模模块，可以创建 8 种类型的 UML 图：活动图、类图、协作图、组件图、部署图、序列图、状态图和用例图，并可以进行正向工程和逆向工程。本章将主要介绍如何在 NetBeans 中进行 UML 建模。

对于不需要学习 UML 建模的读者，可以越过本章继续向后学习。本章相对独立，跳过本章不会对后续的学习产生影响。在学习本章前，建议读者首先学习 UML 的基本知识。本章着重介绍 NetBeans 的使用方法，而不再对 UML 的一些基础知识进行阐述。

17.1 UML 建模支持模块……378
17.1.1 UML 建模模块的功能……378
17.1.2 下载和安装 UML 建模模块……379
17.2 创建 UML 项目以及 UML 建模环境……380
17.2.1 创建 UML 项目……380
17.2.2 “文档”窗口……382
17.2.3 “属性”窗口……383
17.2.4 UML 建模组件面板……383
17.3 UML 图编辑器……384
17.3.1 添加图元素……384
17.3.2 复制、剪切和粘贴元素……384
17.3.3 删除元素……385
17.3.4 锁定编辑……385
17.3.5 选择所有类似的元素……385

Download at http://www.pin5i.com/

17.3.6 调整元素大小……386
17.3.7 堆叠图元素……386
17.3.8 删除所有拐点……386
17.4 UML 图形的缩放……387
17.4.1 放大和缩小……387
17.4.2 缩放到适当大小……388
17.4.3 基于选择区域缩放……388
17.4.4 交互缩放……388
17.5 图布局……388
17.5.1 分层结构布局……388
17.5.2 对称结构布局……388
17.5.3 正交结构布局……389
17.5.4 增量式布局……389
17.5.5 指定布局属性……390
17.6 UML 建模实例……393
17.6.1 创建用例图实例……394
17.6.2 创建类图实例……395
17.7 自动生成代码和反向工程……397
17.7.1 自动生成代码……397
17.7.2 对单个源文件进行逆向工程……399
17.7.3 对项目进行逆向工程……400
17.7.4 使用向导进行逆向工程……401
17.8 UML 建模的其他高级操作……402
17.8.1 查找和替换……403
17.8.2 关联元素……403
17.8.3 过滤元素类型……405
17.8.4 UML 模型报告……405
17.8.5 指定显示和绘制首选项……406
17.9 小结……407

第 18 章 Web 服务以及 SOA 程序开发 408

Web 服务是分布式应用程序组件，它们符合一定的标准，因而可供外部使用。Web 服务可以独立开发，并在不同的系统中运行，因此可以解决各种计算机应用程序的集成问题。面向服务的体系结构 SOA（Service-Oriented Architecture）主要用于组装 Java 业务集成 JBI（Java Business Integration）服务器的部署单元。在复合应用程序项目中，可以使用多种项目类型（例如 BPEL、XSLT 等）组装应用程序。

本章首先介绍 Web 服务的开发方法。然后通过实例介绍了 SOA 程序的简单方法，其中包括如何进行可视化的 BPEL 商业流程开发等内容。

18.1 Web 服务开发……408
18.1.1 Web 服务的类型以及规范……408
18.1.2 创建一个简单的 Web 服务……409
18.1.3 在客户端调用 Web 服务……412

Download at http://www.pin5i.com/

18.2 创建 BPEL 模块项目 ····414
18.3 创建 WSDL 文件 ····415
18.3.1 为 WSDL 创建 XML 架构文件 ····415
18.3.2 创建 WSDL 文件 ····416
18.3.3 可视化的 WSDL 编辑器 ····419
18.4 开发 BPEL 流程 ····421
18.4.1 创建 BPEL 流程 ····421
18.4.2 BPEL 流程编辑器 ····421
18.4.3 BPEL 流程元素 ····422
18.4.4 简单的 BPEL 流程实例 ····423
18.5 开发 SOA 复合应用程序 ····427
18.5.1 创建 SOA 复合应用程序 ····427
18.5.2 创建 Web 客户端测试 SOA 程序 ····428
18.6 小结 ····429

第三篇 NetBeans 手机开发

第 19 章 创建和运行 Java ME 应用程序 432

Java ME 平台是针对小型设备（如寻呼机、移动电话和机顶盒）开发的一组技术和规范。Java ME 定义了许多专门针对电子消费设备和嵌入式设备的 API。在 NetBeans 集成开发环境中通过安装 Mobility Pack 软件包，可以快速开发出 Java ME 应用程序。

本章主要介绍 Mobility Pack 软件包，讲解摩托罗拉 Java ME 开发包的安装和使用，简单 Java ME 应用程序的创建和运行，以及针对 Java ME 应用程序的各种配置。

19.1 Mobility Pack 软件包简介 ····432
19.2 安装 Java ME 开发包 ····433
19.2.1 安装摩托罗拉的 Java ME 开发包 ····433
19.2.2 在 NetBeans 中集成摩托罗拉 Java ME 开发包 ····434
19.2.3 Sun Java Wireless Toolkit 2.5 for CLDC 简介 ····436
19.3 创建一个简单的 Java ME 应用程序 ····437
19.3.1 创建 Mobility 项目 ····437
19.3.2 创建 MIDlet ····439
19.4 Java ME 应用程序结构分析 ····441
19.5 配置 Java ME 应用程序 ····442
19.5.1 查看项目常规属性 ····442
19.5.2 定制平台 ····443
19.5.3 设置应用程序描述符属性 ····443
19.5.4 设置 MIDlet ····445
19.5.5 设置推注册 ····445
19.5.6 设置 API 权限 ····446
19.5.7 过滤源文件 ····447
19.5.8 设置编译属性 ····448

Download at http://www.pin5i.com/

19.5.9 设置库和资源……448
19.5.10 设置 JAR 和 JAD 文件……449
19.5.11 设置 Javadoc 生成……449
19.5.12 设置运行属性……450
19.5.13 设置部署方法……450
19.6 小结……455

第 20 章 Java ME 程序的可视化开发 456

NetBeans Mobility Pack 中提供了可视化的 Mobile 设计器，利用其提供的强大功能可以通过可视方式来设计应用程序流和图形用户界面 GUI。在将对象拖放到流设计器和屏幕设计器后，NetBeans 将自动生成并更新实现设计的 Java 代码。这样，开发人员可以将更多的精力集中于业务逻辑上，从而大大提高了开发效率。

本章首先简要介绍简单的可视 MIDlet 的创建方法，以及可视化的 Mobile 设计器的使用方法。然后，通过开发实例具体讲解每个 MIDP 组件的使用方法，特别是对 Mobility Pack 提供的 SplashScreen、WaitScreen、FileBrowser、TableItem 等组件以及各种 SVG 组件进行了详细的介绍。

20.1 创建可视的 MIDlet……456
20.1.1 创建可视化的 MIDlet……457
20.1.2 可视化 MIDlet 组件面板……458
20.1.3 流设计器……458
20.1.4 屏幕设计器……459
20.1.5 “导航”窗口……460
20.1.6 “属性”窗口……460
20.2 MIDP 2.0 高级界面组件……460
20.3 创建基于文本框（TextBox）的应用程序……461
20.3.1 创建文本框应用程序……461
20.3.2 TextBox 的常用属性……462
20.3.3 使用命令按钮……464
20.4 创建闪屏（SplashScreen）……466
20.5 使用滚动消息栏（Ticker）……469
20.6 创建警报（Alert）……470
20.7 创建等待屏幕（WaitScreen）……473
20.8 创建登录屏幕（LoginScreen）……476
20.9 创建 SMS 编写器（SMSComposer）……479
20.10 创建列表（List）……481
20.11 创建文件浏览器……483
20.12 创建 PIM 浏览器（PIMBrowser）……485
20.13 创建表单（Form）……486
20.13.1 使用文本字段组件（TextField）……487
20.13.2 使用字符串项组件（StringItem）……489
20.13.3 使用图像项组件（ImageItem）……491
20.13.4 使用标尺组件（Gauge）……493

Download at http://www.pin5i.com/

20.13.5 使用日期字段组件（DateField）……496
20.13.6 使用选项组组件（ChoiceGroup）……497
20.13.7 使用 TableItem 组件……498
20.13.8 使用分隔符组件（Spacer）……501
20.14 使用 SVG 组件……502
20.14.1 预览 SVG 文件……503
20.14.2 设置外部编辑器……504
20.14.3 导出 SVG 图像和动画……504
20.14.4 使用 SVG 播放器组件（SVGPlayer）……506
20.14.5 使用 SVG 闪屏（SVGSplashScreen 组件）和 SVG 等待屏幕（SVGWaitScreen 组件）……507
20.14.6 使用 SVG 菜单组件（SVGMenu）……508
20.15 使用组件面板管理器……510
20.16 使用 Mobile Game Builder……511
20.16.1 创建 Game Builder 样例项目……511
20.16.2 创建场景和平铺层……513
20.16.3 创建动画层……515
20.16.4 创建 GameCanvas 和 MIDlet……516
20.17 小结……517

第 21 章 Java ME 开发高级技术 518

前面两章中介绍了在 NetBeans 集成开发环境中开发 Java ME 程序的基本方法，本章将介绍一些高级技术，包括签名运行、进行混淆、进行单元测试等内容。对于初学者，可以跳过本章继续学习后续章节。

21.1 创建 Canvas 画布……518
21.2 应用程序的签名运行……521
21.2.1 签名概述……521
21.2.2 NetBeans 的安全管理器……522
21.2.3 MIDP 项目签名……523
21.2.4 指定安全域运行……524
21.3 对应用程序进行混淆处理……524
21.4 多编译配置以及预编译……525
21.4.1 配置管理……526
21.4.2 使用更多配置……527
21.4.3 使用预编译……528
21.5 小结……530

第四篇 NetBeans 组件开发

第 22 章 开发 NetBeans 插件模块 532

NetBeans 是 Sun 公司领导的一个成功的开发源码项目。前面几章主要介绍了 NetBeans 集

Download at http://www.pin5i.com/

成开发环境的各种使用和开发方法。本章将介绍 NetBeans 项目中另外一个产品：NetBeans 平台。开发人员可以在 NetBeans 平台的基础上进行自己的应用程序开发，从而加快程序的开发过程。

本章首先介绍 NetBeans 集成开发环境和 NetBeans 平台的区别，然后着重介绍如何开发 NetBeans 集成开发环境扩展模块，以及开发基于 NetBeans 平台的富客户端应用程序。本章旨在为 NetBeans 平台开发的读者指引一条入门的道路。对于不打算进行 NetBeans 平台开发的读者，只需要阅读本书的第一节，弄清楚 NetBeans 集成开发环境和 NetBeans 平台的概念即可。

22.1 NetBeans IDE 和 NetBeans 平台……532
22.1.1 NetBeans 平台概述……532
22.1.2 NetBeans 平台的功能……533
22.2 NetBeans 模块开发……533
22.2.1 创建 NetBeans 插件模块项目……533
22.2.2 创建菜单项和工具栏按钮……536
22.2.3 安装和卸载插件模块……540
22.3 插件模块项目管理……541
22.3.1 源和代码级别……541
22.3.2 库管理……542
22.3.3 设置显示参数……544
22.3.4 版本控制……544
22.3.5 打包设置……545
22.4 为操作编写处理代码……546
22.5 富客户端应用程序开发……548
22.5.1 富客户端应用程序开发概述……548
22.5.2 创建模块套件项目……548
22.5.3 添加模块项目……549
22.5.4 生成选项……550
22.5.5 去除不必要的菜单……552
22.5.6 创建闪屏……552
22.5.7 分发应用程序……554
22.6 小结……554

第五篇 NetBeans 开发 C++程序

第 23 章 使用 NetBeans 开发 C/C++程序 556

NetBeans 不仅仅是一个功能强大的 Java 开发工具，也可以成为其他语言的开发平台。如果安装了具有 C/C++支持的 NetBeans 集成开发环境，还可以开发 C 和 C++应用程序，从而使 NetBeans 成为一款跨平台的 C/C++开发工具。带有 C/C++支持的 NetBeans 提供了 C/C++项目和模板、高级源代码编辑器、对编译器的支持以及 Makefile 和配置等功能，使开发 C/C++项目变得轻松快捷。

本章主要介绍了 NetBeans C/C++ Pack 软件包的安装和设置以及 C 和 C++项目的创建，最后着重讲解了利用 NetBeans C/++ Pack 软件包创建 JNI。

23.1 NetBeans C/C++ Pack 软件包简介……556
23.2 安装和设置 NetBeans C/C++ Pack 软件包……556

Download at http://www.pin5i.com/

23.2.1 下载安装具有 C/C++支持的 NetBeans 集成开发环境……557
23.2.2 安装并设置编译器和工具……558
23.3 创建和设置 C 和 C++项目……562
23.3.1 创建 C/C++应用程序……562
23.3.2 逻辑视图和物理视图……563
23.3.3 向项目中添加文件……563
23.3.4 编辑源文件……564
23.3.5 设置项目属性……565
23.3.6 管理配置……567
23.3.7 生成并运行项目……568
23.4 设置 C/C++ Pack……568
23.4.1 设置 C/C++生成工具……569
23.4.2 设置 C/C++项目选项……570
23.4.3 设置 C/C++代码帮助……570
23.5 创建 Makefile……571
23.5.1 Makefile 简介……571
23.5.2 使用 Makefile 向导创建 Makefile……572
23.6 创建基于现有代码的 C/C++项目……579
23.7 利用 C/C++ Pack 创建 JNI……581
23.7.1 JNI 简介……581
23.7.2 创建 Java 应用程序项目……582
23.7.3 创建 C/C++动态库项目……583
23.7.4 生成并运行应用程序……586
23.8 小结……586

附录 A NetBeans 的快捷键 587

Download at http://www.pin5i.com/

PART 1

第一篇　标准开发环境

第 1 章　NetBeans 的安装和开发环境配置
第 2 章　通过 NetBeans 开发 Java 标准应用程序
第 3 章　在 NetBeans 中调试应用程序
第 4 章　代码重构
第 5 章　JavaBean 开发
第 6 章　管理和使用服务器
第 7 章　数据库的管理与开发

Download at http://www.pin5i.com/

第1章

NetBeans的安装和开发环境配置

NetBeans是Sun公司使用Java语言开发的一个开源工具，主要包括IDE（集成开发环境）和Platform（平台）两个主要部分。其中，集成开发环境是在平台基础上实现的，并且平台本身也开放给开发人员直接使用。本书的大部分内容介绍NetBeans集成开发环境的使用方法，只在第5篇中介绍NetBeans平台的开发使用方法。

利用NetBeans集成开发环境可以开发标准的Java应用程序、Web程序、手机程序、C++程序等。本章首先介绍NetBeans集成开发环境的安装和基本结构，以及如何更新NetBeans等内容。

本章主要内容

- NetBeans的各个菜单功能
- NetBeans的各个窗口
- 源代码编辑器
- NetBeans的模块安装/更新方法

本章亮点

- NetBeans的安装
- NetBeans模块的手工安装方法
- NetBeans源代码编辑器的使用

1.1 下载和安装NetBeans

NetBeans IDE（集成开发环境）是一个免费的、开源的集成开发环境。它可以运行在众多常见的平台上，包括Windows、Linux、Solaris和MacOS等。NetBeans集成开发环境为开发者提供了所需的所有工具，使得开发专业的跨平台的桌面、企业、Web和移动应用十分简便。

目前NetBeans的最新版本为NetBeans IDE 6.0，因此本书以NetBeans 6.0简体中文版为目标进行介绍。首先介绍NetBeans 6.0的新特性，安装NetBeans 6.0的最低系统要求，然后介绍如何在NetBeans的官方网站下载NetBeans 6.0，最后给出具体的安装步骤。

1.1.1 NetBeans 6.0新功能及改进

NetBeans 6.0是在NetBeans 5.5之后的一个新版本，在继承了NetBeans 5.5优秀功能基础上，NetBeans 6.0又提供了如下的功能及改进。

1．功能强大的代码编辑器

语言敏感的编辑器支持缩进、自动补全和高亮显示。代码编辑器可以自动分析代码、自动匹配单词和括号、标注代码错误、显示提示和Javadoc，如图1-1所示。编辑器还可以被定制化或者横向纵向切分，同时提供集成的代码重构、调试和JUnit测试。

Download at http://www.pin5i.com/

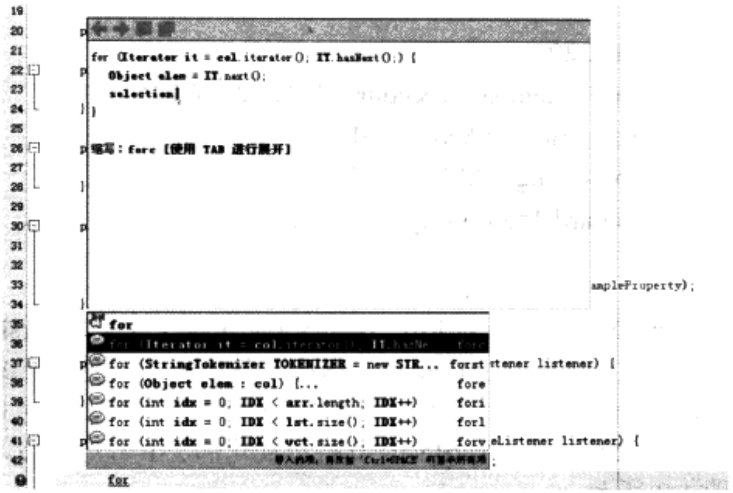


图 1-1 NetBeans 的编辑器

2. 增强的图形用户界面编辑器（GUI Builder）

NetBeans 6.0 中可以通过拖曳设计基于 Swing 组件的图形用户界面，只需要从组建面板里选择组件并拖曳到 GUI Builder 里，再双击就可以直接编辑 JLabel、JButton 和 JTextField 的文字属性，如图 1-2 所示。在属性编辑器窗口中，也可以对选中的组件属性进行编辑。NetBeans 集成开发环境内建有对本地化和国际化的支持，可以开发多个语言的应用程序。

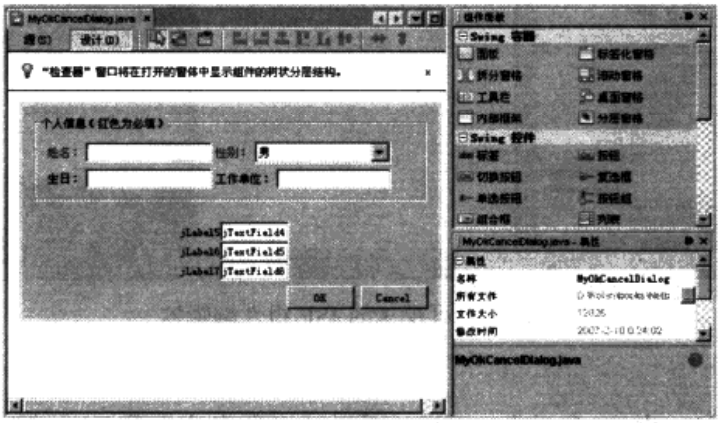


图 1-2 用户界面编辑器

3. 全面支持 Java 企业版应用程序

NetBeans 6.0 全面支持 Glassfish、Sun Java System Application Server PE 9、Weblogic 9.0、JBoss 4 以及 Tomcat 服务器等，同时支持 J2EE 1.4 和 Java EE 5，包括 JAX-RPC 和 JSR 109 标准。

NetBeans 6.0 完全支持 Java EE 5 的集成开发环境，而且增强了 Java Persistence、EJB 3 和 JAX-WS 2.0 规格说明。在下载安装 NetBeans 6.0 后，任何有关 Java EE 5 的特性都可以正常使用，其中包括 EJB 3.0 和 Web Services 等。NetBeans 6.0 提供了对 Java EE 5 API 的代码自动完成和 Javadoc 功能支持。NetBeans 6.0 支持如下的一些新技术：

- Servlet 2.5
- JavaServer Pages 2.1
- JavaServer Faces 1.2（JSR-127）
- Struts 1.2.9
- Enterprise JavaBeans（EJB）3.0

Download at http://www.pin5i.com/

- JAX-WS 2.0
- Java Persistence（JSR-220）
- Java API for XML-based Remote Procedure Calls（JAX-RPC）1.6
- Java API for XML Registries（JAXR）1.0
- Java API for XML Processing（JAXP）1.2
- JavaServer Pages Standard Tag Library 1.1

4. Web 程序开发

NetBeans 5.5 支持 JavaServer Faces（JSF）、JavaServer Pages（JSP）、Persistence Units、Struts、AJAX 和 JavaServer Pages Standard Tag Library（JSTL）等技术，同时提供编辑部署描述符可视化的编辑器和调试 Web 应用的 HTTP 监视器。只需要几个简单的步骤就可以创建，从数据库中提取数据拥有完全功能的 JSF 页面。NetBeans 支持可视化的 JSF 程序开发，使得开发页面可以通过拖曳完成，如图 1-3 所示。

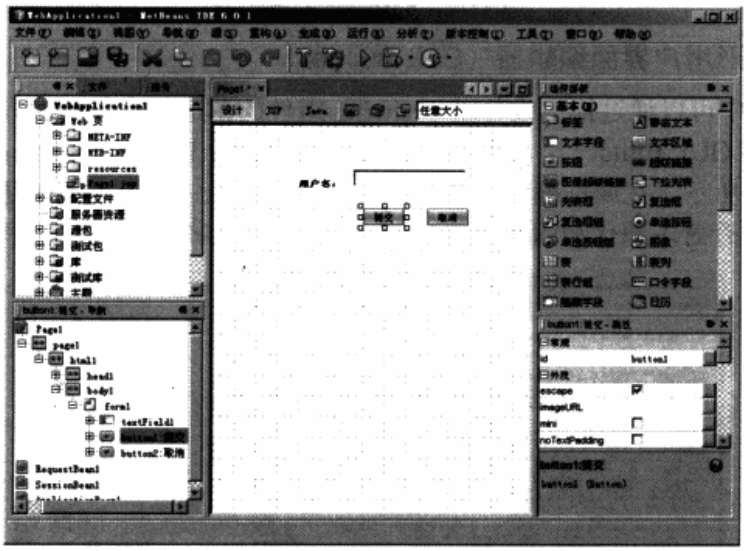


图 1-3 可视化的 JSF 组件编辑器

5. 协同开发

使用 NetBeans Developer Collaboration（可以从更新中心下载），开发人员可以通过网络实时地共享整个项目和文件。这样就可以让其他的同事在远程进行修改，而您可以在本地看着它键入字符，然后运行您的应用。协同开发还可以检查其他伙伴的工作并发送消息。而这消息可以是普通文本，或者有高亮支持的 XML、HTML 和 Java 代码。

6. Ruby 和 Rails 支持

从 NetBeans 6.0 开始加入了对 Ruby 开发的支持，提供了功能强大的 Ruby 编辑器（具有代码自动完成功能）和调试器，并全面支持 Rails。在 NetBeans 6.0 中可以创建 Ruby 项目，在项目中支持 Ruby 文件、RSpect 规范文件、YAML 文件。开发人员可以选择使用 JRuby 运行时或者其他的任何本地 Ruby 运行时。

7. 对版本控制 CVS 和 Subversion 的支持

版本控制已经被紧密地集成到 NetBeans 集成开发环境的工作流当中，如图 1-4 所示。NetBeans 可以自动识别 CVS 工作目录。开发人员可以选择使用内建的 CVS，或者从更新中心下载 Subversion 模块。

Download at http://www.pin5i.com/

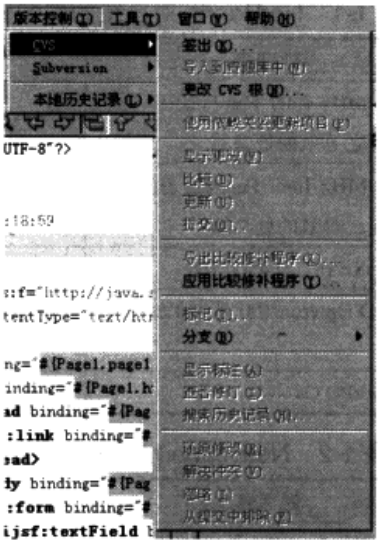


图 1-4 NetBeans 紧密集成版本控制

8. 可视化的手机程序开发

NetBeans 6.0 提供了可视化的手机程序开发支持，利用其提供的可视化设计器，可以通过拖曳的方式往手机界面上添加组件。在 NetBeans 6.0 开始加入了一个可视化的游戏编辑器，通过该编辑器可以使用 MIDP 2.0 中的游戏编程接口来快速地设计游戏。在这个设计器中，允许用户创建场景、游戏精灵、游戏图层，以及进行游戏图层的管理，如图 1-5 所示。

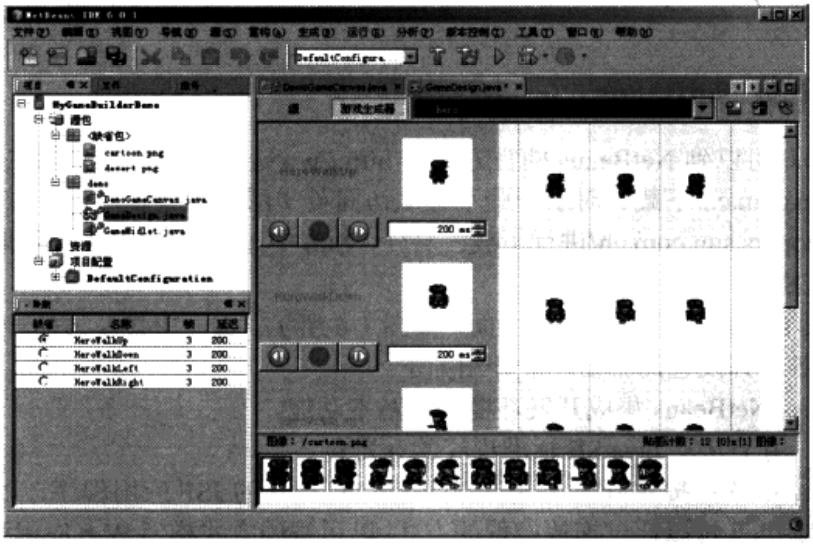


图 1-5 可视化的游戏编辑器

1.1.2 NetBeans 对系统的最低要求

NetBeans 可以在不同的系统上运行，在下载安装 NetBeans 前需要清楚 NetBeans 对系统的最低要求以及推荐的系统配置。表 1-1 给出了 NetBeans 在各个系统中的最低要求。表 1-2 给出了 NetBeans 推荐的硬件配置。

说明：NetBeans 集成开发环境要求的最低屏幕分辨率为 1024×768 像素。

Download at http://www.pin5i.com/

表 1-1 NetBeans 对系统的最低要求

操作系统	处理器	内 存	硬盘空间
Microsoft Windows 操作系统	500 MHz Intel Pentium III 工作站或与之等效的处理器	512 MB	850 MB 可用硬盘空间
Linux 操作系统	500 MHz Intel Pentium III 工作站或与之等效的处理器	512 MB	450 MB 可用硬盘空间
Solaris OS（SPARC）	UltraSPARC II 450 MHz	512 MB	450 MB 可用硬盘空间
Solaris OS（x86/x64 Platform Edition）	AMD Opteron 100 系列 1.8 GHz	512 MB	450 MB 可用硬盘空间
Macintosh OS X 操作系统	PowerPC G4	512 MB	450 MB 可用硬盘空间

表 1-2 NetBeans 推荐的硬件配置

操作系统	处理器	内 存	硬盘空间
Microsoft Windows 操作系统	1.4 GHz Intel Pentium III 工作站或与之等效的处理器	1 GB	1 GB 可用硬盘空间
Linux 操作系统	1.4 GHz Intel Pentium III 工作站或与之等效的处理器	1 GB	850 MB 可用硬盘空间
Solaris OS（SPARC）	UltraSPARC IIIi 1 GHz	1 GB	450 MB 可用硬盘空间
Solaris OS（x86/x64 Platform Edition）	AMD Opteron 100 系列 1.8 GHz	1 GB	450 MB 可用硬盘空间
Macintosh OS X 操作系统	PowerPC G5	1 GB	450 MB 可用硬盘空间



1.1.3 下载 NetBeans 6.0

NetBeans 6.0 可以到 NetBeans 的官方网站 http://www.netbeans.org 或者 Sun 中国技术社区 http://gceclub.sun.com.cn/下载。对于一些不喜欢访问英文网站的读者，也可以到 Sun 中国技术社区 http://developers.sun.com.cn/进行下载。另外，Sun 中国技术社区中还提供了一些中文的学习资料和参考文档。

NetBeans 的官方网站为英文操作界面，如图 1-6 所示。页面中最左边的一列表示 NetBeans 6.0 的开发包，各个开发包的主要功能介绍如下。

- 基本 IDE：NetBeans 集成开发环境的最基本功能。
- Java SE：用于开发 Java SE 标准程序的开发包。
- Web & Java EE：提供开发 Web 应用程序、可视化的 JSF 应用程序，以及 Java EE 企业级程序的功能。可以通过拖放控件的方式，以可视的方式构造 Web 应用程序。在 NetBeans 集成开发环境中添加了以可视方式快速生成基于标准的 Web 应用程序所需的所有功能，其中包括 AJAX 和 JSF 组件支持。
- Mobility：可以开发基于 CLDC/MIDP 或者 CDC 的手机应用程序，使用可视工具快速生成手机程序的图形界面。具体开发包的功能，读者可以参考第 19.1 节的介绍。
- UML：提供了创建 UML 项目的功能，可以进行 UML 双向建模。
- SOA：加入了开发专业 SOA 应用的所有功能。开发人员可以用 XML、BPEL 和 Java Web 服务编写、测试和调试 SOA 应用。
- Ruby：提供了开发 Ruby 应用程序的功能，提供了功能强大的 Ruby 编辑器、调试器，并全面支持 Rails。

Download at http://www.pin5i.com/

- C/C++：提供了在 NetBeans 集成开发环境中创建 C/C++应用程序的能力。可以为很多不同的平台编写 C 和 C++的代码，包括 Microsoft Windows、Linux 和 Solaris。



说明：由于网站的页面和链接有可能随时更换，本书给出的链接和页面不能保证在任何时候都与网站一致，因此读者应该以网站为准。本书仅作为参考。

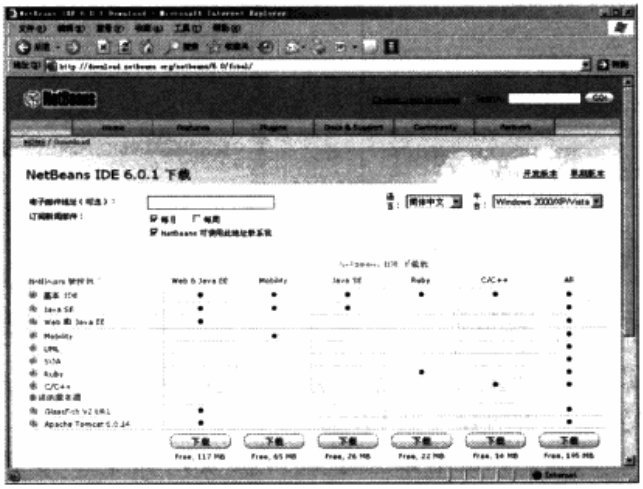


图 1-6 从 NetBeans 官方网站下载 NetBeans

页面的各个列表示的是包含不同功能的安装包，其中各个安装包都默认包含了 NetBeans 集成开发环境的最基本功能（基本 IDE），安装包的详细说明如表 1-3 所示。



说明：这里建议中国用户使用简体中文版的 NetBeans。在目前比较流行的几个开发工具（例如 JBuilder、Eclipse）中，NetBeans 是汉化得最好的。不但对其界面进行了无缝的汉化，而且连开发文档、技术手册等也都是中文的。

表 1-3 NetBeans 6.0 安装包的详细说明

安装包	大 小	说 明
NetBeans IDE 6.0 Web 和 Java EE 版本	96MB	包含 Java SE 和 Web & Java EE 开发包，即可以进行 Java SE、Web 程序和 Java EE 的开发，并包含 GlassFish V2 服务器和 Tomcat 服务器
NetBeans IDE 6.0 移动开发版本	58MB	包含 Java SE 和 Mobility 开发包，即可以进行 Java SE、手机移动程序的开发
NetBeans IDE 6.0 Java SE 版本	21MB	包含 Java SE 开发包，只能进行 Java SE 应用程序的开发
NetBeans IDE 6.0 Ruby 版本	19MB	包含 Ruby 开发包，只能进行 Ruby 程序的开发
NetBeans IDE 6.0 C/C++版本	11MB	包含 C/C++开发包，只能进行 C/C++应用程序的开发
NetBeans IDE 6.0 全功能版本	169MB	包含以上版本的全部功能，本书就是以这个版本为目标来进行介绍的



注意：Mobility Web 服务连接向导仅包含在 All 下载中，并不包含在 Mobility 下载中。



1.1.4 安装 NetBeans 6.0 全功能包

Netbeans 可以在 Windows、OS2、OpenVMS 和 Linux 等平台上运行，本节以 Windows 平台为目标平台，介绍 NetBeans IDE 6.0 全功能版本的安装方法和过程。

Download at http://www.pin5i.com/

在下载 NetBeans 安装文件后，双击该文件开始安装，显示如图 1-7 所示的对话框。在这个对话框中显示了欢迎信息和要安装的 NetBeans 的开发包。单击“定制”按钮，可以打开如图 1-8 所示的“定制安装”对话框。

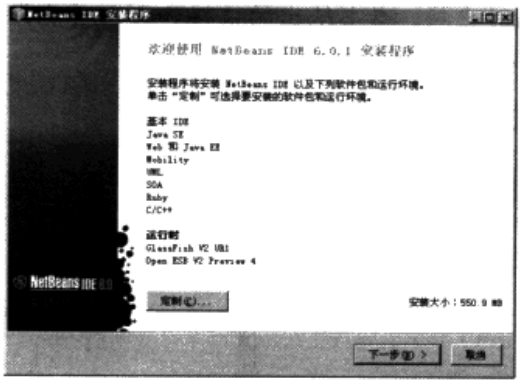


图 1-7 显示安装的欢迎信息

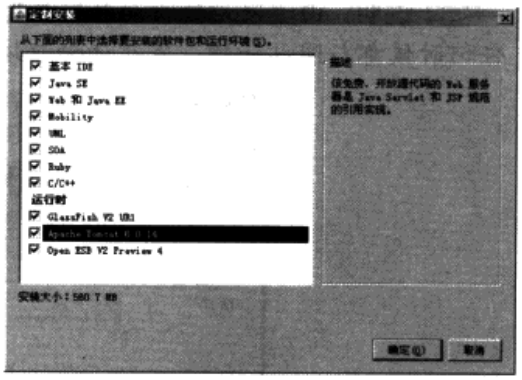


图 1-8 “定制安装”对话框

这里请选中所有的功能，包括 Tomcat 服务器。返回欢迎界面后，单击“下一步”按钮，打开如图 1-9 所示的“安装许可协议”对话框。

“安装许可协议”对话框主要用来显示许可协议，如果不接受许可协议将不能安装 NetBeans。这里选中“我接受许可协议中的条款”单选按钮，再单击“下一步”按钮打开如图 1-10 所示的“安装路径选择”对话框。

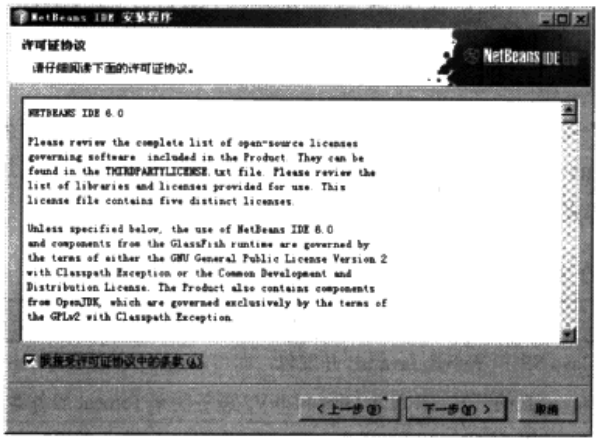


图 1-9 “安装许可协议”对话框

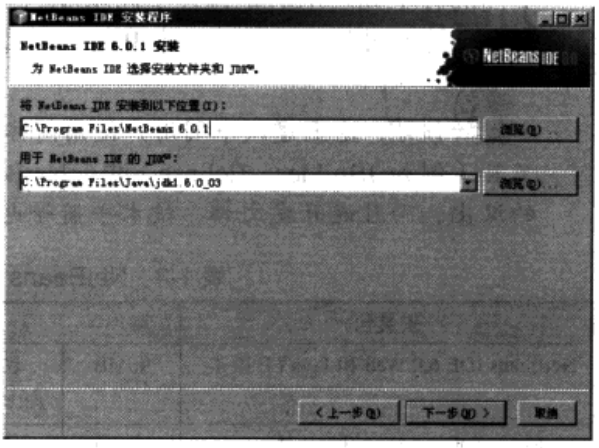


图 1-10 “安装路径选择”对话框

“安装路径选择”对话框主要用于设置 NetBeans 的安装路径，这里使用默认的安装目录。如果要安装到其他的目录，可以单击“浏览”按钮进行选择。如果系统中安装了多个 JDK，那么可以在“选择 JDK”对话框中选择 NetBeans 要运行在哪个 JDK 之上。



注意：NetBeans 的运行需要 JDK 5.0 或者以上版本，因此在安装 NetBeans 前请先下载安装 JDK。这里强烈建议读者安装 JDK 6.0 或者以上版本。

选择完毕，单击“下一步”按钮打开“GlassFish V2 UR1 安装”对话框，如图 1-11 所示。在“将 GlasshFish 安装到”文本框中输入 GlassFish V2 服务器的安装路径。在“用于 GlassFish 应用服务器的 JDK”下拉列表框中选择 GlassFish 应用服务器使用的 JDK 版本。“管理员用户名”文本框中为服务器的管理员账户，默认值为 admin。下面的两个文本框为管理员账号的密码，默认值为 adminadmin。“HTTP 端口”文本框用于指定使用 HTTP 协议访问服务器时使用

Download at http://www.pin5i.com/

的端口号，默认值为 8080。“HTTPS 端口”文本框用于指定使用 HTTPS 协议访问服务器时使用的端口号，默认值为 8081。“管理端口”文本框用于指定使用管理控制台对服务器进行管理时使用的端口号，默认值为 4848。

这里，接受所有的默认值，单击“下一步”按钮，打开安装 Tomcat 服务器的对话框，如图 1-12 所示。该对话框只有在如图 1-8 所示的“定制安装”对话框中选择安装 Tomcat 服务器后才会出现。

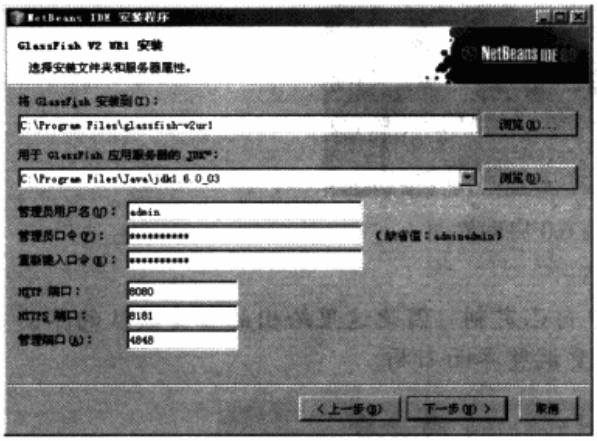


图 1-11 “GlassFish V2 UR1 安装”对话框

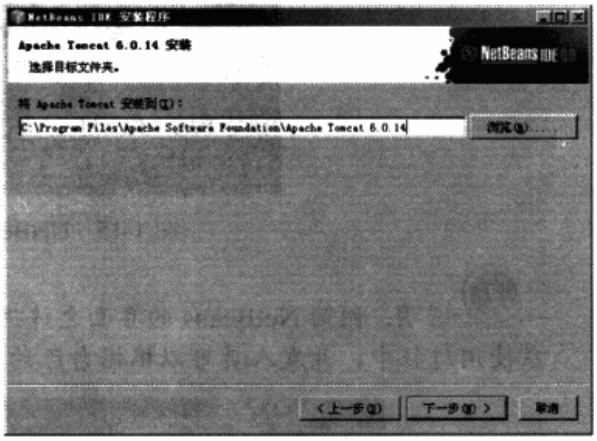


图 1-12 安装 Tomcat 服务器

接受默认的设置，单击“下一步”按钮打开“摘要”对话框，如图 1-13 所示。

确认安装信息无误后，单击“安装”按钮开始安装。安装完毕后，显示如图 1-14 所示的安装成功对话框。单击“完成”按钮结束安装，安装程序自动创建程序组并在桌面创建一个 NetBeans 的图标。

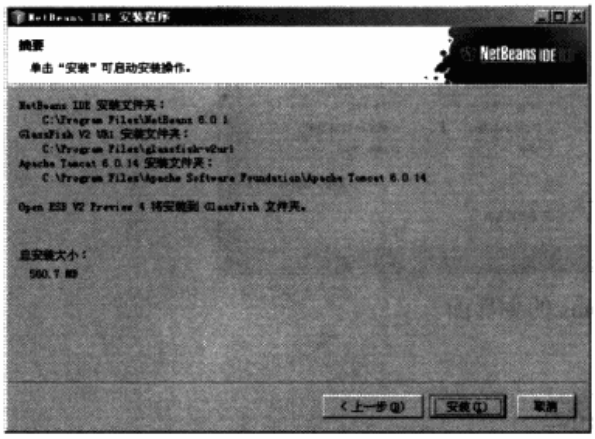


图 1-13 “摘要”对话框

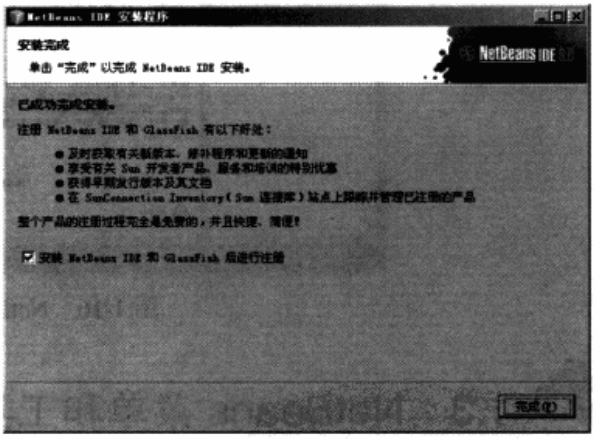


图 1-14 安装成功对话框

1.2 NetBeans 集成开发环境

运行 NetBeans 6.0 集成开发环境，显示如图 1-15 所示的启动界面。NetBeans 启动后的界面如图 1-16 所示，可以看到界面主要由菜单、工具栏、代码编辑区和其他窗口组成。

在 NetBeans 主界面中，最上面的是菜单栏，菜单栏下面是工具栏，最下面是 NetBeans 的各个窗口。其中，左下角的是欢迎窗口，在第一次运行 NetBeans 时都会默认打开欢迎窗口。在欢迎窗口中，开发人员可以快速地打开项目，访问 NetBeans 的开发文档和帮助文档、在线

Download at http://www.pin5i.com/

浏览文章和新闻，以及浏览 NetBeans 的博客。如果不希望每次启动的时候都打开欢迎窗口，那么可以取消选中的“启动时显示”复选框。

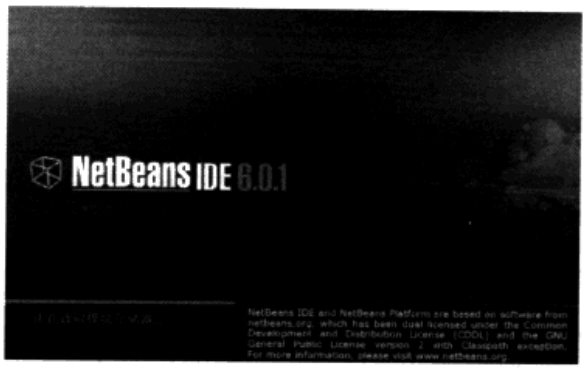


图 1-15 NetBeans 6.0 启动界面



说明：因为 NetBeans 的界面允许用户自己定制，因此这里给出的都是默认的界面。在使用过程中，开发人员可以根据自己的需要调整界面布局。

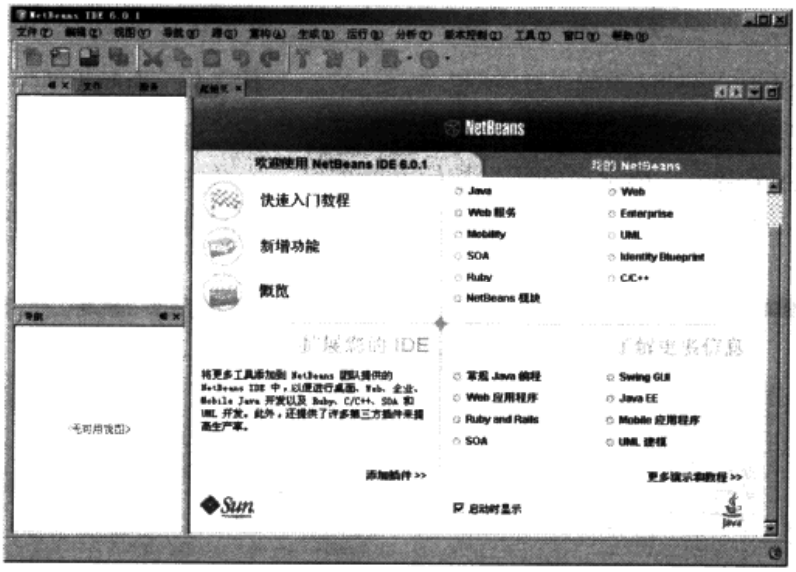


图 1-16 NetBeans 的主界面



1.3 NetBeans 菜单和工具栏

1. NetBeans 菜单栏

NetBeans 菜单栏如图 1-17 所示，包括了文件、编辑、视图、导航、源、重构、生成、运行、分析、版本控制、工具、窗口和帮助等菜单。



图 1-17 NetBeans 菜单栏

下面介绍各个菜单的主要功能。

- “文件”菜单：包含了文件和项目的一些基本操作命令，例如打开项目、打开文件、关闭项目、打开最近的项目、设置主项目、设置项目属性等。

Download at http://www.pin5i.com/

- “编辑”菜单：包括了各种简单的操作（例如复制、粘贴、剪切等），以及查找、替换功能和跳转命令（例如跳转到某一行、跳转到声明等）。
- “视图”菜单：包括了各种视图的操作，例如进行代码折叠、显示工具栏、显示行号等。
- “导航”菜单：主要提供了在代码中进行跳转的各种功能，例如从一个方法中跳转到另外一个方法中，或者跳转到下一个书签或者错误的行等。
- “源”菜单：对源代码的一些操作和控制，例如对代码进行格式化，使得代码符合一定的缩进规则；插入某些代码、修复一些引用错误等。
- “重构”菜单：重构菜单中提供了对代码重构的一些操作，例如重命名、更改方法参数、封装字段、移动类等。
- “生成”菜单：主要用于编辑代码或者生成项目。
- “运行”菜单：运行菜单中提供了各种运行和调试的命令，例如运行主项目、单步调试、设置断点等内容。
- “分析”菜单：用来对内存使用情况或者程序运行的性能进行分析，以帮助开发人员对系统进行优化。
- “版本控制”菜单：提供了版本控制的功能，该功能对于团队开发非常的实用。可以使用的版本控制有 CVS、PVCS、Subversion 等。
- “工具”菜单：提供了各种管理工具，例如组件面板管理器、Java 平台管理器、库管理器、更新中心等。
- “窗口”菜单：提供了打开和关闭各种窗口的操作，后面介绍的几个窗口都可以在这个菜单中打开。
- “帮助”：提供了 NetBeans 的帮助内容、欢迎屏幕、关于等内容。

2. NetBeans 工具栏

NetBeans 的工具栏如图 1-18 所示，在工具栏上提供了一些常用的命令，例如打开项目、复制、运行等。把鼠标停留在工具栏的某个按钮上时，会显示该按钮功能的提示信息以及快捷键。对于 NetBeans 5.5 简体中文版，这些提示信息也都是中文的。



图 1-18　NetBeans 的工具栏

NetBeans 的工具栏可以进行定制，在工具栏上单击右键，会显示如图 1-19 所示的上下文菜单。在这个弹出菜单中用户可以选择哪个工具条显示，哪个隐藏。值得一提的是，用户可以通过选择“内存”命令打开“内存”工具条，如图 1-20 所示。



图 1-19　工具栏上下文菜单



图 1-20　“内存”工具条

“内存”工具条显示了当前内存的使用情况，单击它会强制执行 Java 的垃圾回收，以释放更多的内存空间。

选择工具栏上下文菜单“定制...”命令，会打开如图 1-21 所示的“定制工具栏”对话框。用户可以在这个对话框中修改按钮的顺序、更改工具栏的名字、添加新工具栏等。

Download at http://www.pin5i.com/



图 1-21 “定制工具栏”对话框

1.4 NetBeans 窗口

NetBeans 集成开发环境的主要组成部分是实现不同功能的窗口。本节将介绍几个常用的窗口，包括“项目”窗口、“服务”窗口、“文件”窗口、“输出”窗口、“导航”窗口等。其他的窗口将在其他相关章节进行介绍。

1.4.1 “项目”窗口

“项目”窗口列出了当前打开的所有项目，是项目源的主入口点。展开某个项目会看到项目中使用的项目内容的逻辑视图。

- 源包：包括源代码包中的源代码文件，如图 1-22 所示。双击某个源文件节点可以打开该文件进行编辑。
- 测试包：编写的单元测试代码将会包含在这个包中。
- 库：项目使用的库文件。
- 测试库：编写的测试程序使用的测试库。

虽然在 NetBeans 中可以同时打开多个项目，但是同一时间只能有一个主项目，在“项目”窗口中可以进行主项目的设置。

“项目”窗口中的每个节点都有一个弹出式菜单，其中包含可以对该节点运行的命令。通过右键单击某个节点，可以打开该节点的弹出式菜单。

项目节点的弹出式菜单中包含所有主要项目命令，如图 1-23 所示，例如“生成”、“运行”和“调试”命令等。项目节点的弹出式菜单中还包含“属性”命令，可以使用该命令来配置项目。

说明：默认情况下，将在集成开发环境的左上角打开“项目”窗口。也可以通过选择“窗口”→“项目”命令或者快捷键 Ctrl+1 打开“项目”窗口。

表 1-4 给出了“项目”窗口中使用的图标以及图标代表的内容。

Download at http://www.pin5i.com/



图 1-22 “项目”窗口

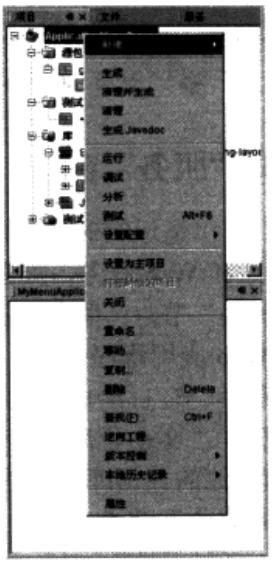


图 1-23 项目节点的弹出式菜单

表 1-4 “项目”窗口中的图标

图 标	描 述	图 标	描 述
	标准 Java 项目		企业应用程序客户端项目
	标准 Web 项目		打包的企业归档项目
	EJB 模块项目		自由格式项目
	企业应用程序项目		Web 页节点。此节点出现在 Web 项目中
	NetBeans 插件模块项目		Java 包根目录。在自由格式项目中，每个源文件夹都有一个 Java 包根目录节点
	Java 包		私有 Java 包
	空 Java 包		公共 Java 包

1.4.2 “文件”窗口

“文件”窗口和“项目”窗口有些类似，用于显示基于目录的项目视图，其中包括“项目”窗口中未显示的文件和文件夹，如图 1-24 所示。

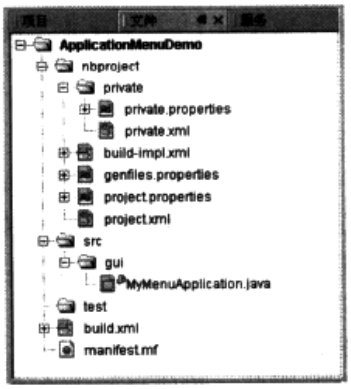


图 1-24 “文件”窗口

“文件”窗口中的每个节点也都有一个弹出式菜单，其中包含可以对该节点运行的命令，例如复制、删除、查找等。通过右键单击某个节点，可以打开该节点的弹出式菜单。

Download at http://www.pin5i.com/

说明：默认情况下，将在集成开发环境的左上角打开“文件”窗口。也可以通过选择“窗口”→“文件”命令或者快捷键 Ctrl+2 打开“文件”窗口。

1.4.3 “服务”窗口

“服务”窗口是运行时资源的主入口点。它显示了资源的逻辑视图，如在 IDE 中注册的服务器、数据库和 Web 服务，如图 1-25 所示。“服务”窗口中各个节点的含义介绍如下。

- “服务器”节点：服务器节点给出了注册的所有服务器，例如 Tomcat 服务器可用于开发 JSP Web 应用程序。
- “数据库”节点：给出了支持的数据库驱动程序，图 1-25 中显示的是 ODBC-JDBC Bridge。右键单击“驱动程序”节点，在弹出的菜单中选择“添加驱动程序”命令，可以为 NetBeans 添加其他的数据库驱动程序。
- “Web 服务”节点：用于管理所有的 Web 服务。

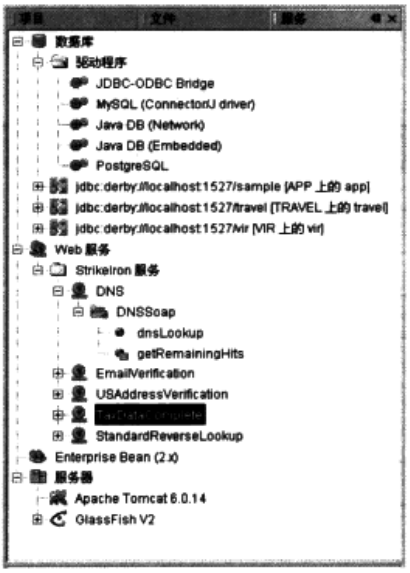


图 1-25 “服务”窗口

说明：默认情况下，将在集成开发环境的左上角打开“服务”窗口。也可以通过选择“窗口”→“服务”命令或者快捷键 Ctrl+5 打开“服务”窗口。

1.4.4 “输出”窗口

“输出”窗口是一个多标签窗口，可显示来自集成开发环境的消息。在出现编译错误、调试程序、生成 Javadoc 文档等情况下，将自动显示此窗口，如图 1-26 所示。输出窗口在运行和调试时非常有用，运行窗口中不仅显示了程序的运行信息，还能够显示通过 System.out.print 方法打印的信息以及一些异常信息。

在运行需要用户输入的程序时，“输出”窗口中将显示一个新标签，并且光标将停留在此标签中。可以在“输出”窗口中输入信息，此信息与在命令行中输入的信息一样。

在编译程序时，如果程序中有错误，错误消息显示为带下划线的蓝色文本，并链接到产生错误的源代码行上，如图 1-27 所示。单击该链接可以快速地定位到发生错误的代码行。

Download at http://www.pin5i.com/

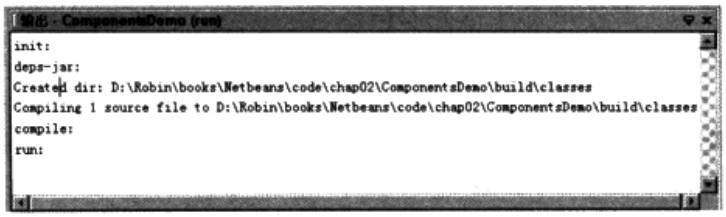


图 1-26 “输出”窗口

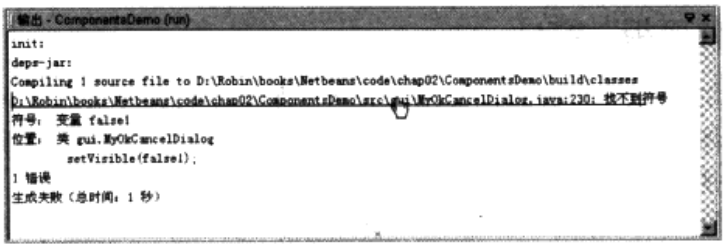


图 1-27 显示编译错误



说明：可以通过选择“窗口”→“输出”命令或者快捷键 Ctrl+4 打开“输出”窗口。



1.4.5 “导航”窗口

“导航”窗口提供了当前选定文件的简洁视图，并且简化了文件不同部分之间的导航。如果在源代码编辑器中打开某个文件，那么会在“导航”窗口中显示构造函数、方法和字段，如图 1-28 所示。

通过将鼠标光标停在成员的节点上，可以为成员显示 Javadoc 文档。在“导航”窗口中双击某个成员可以在源代码编辑器中快速地定位该成员。

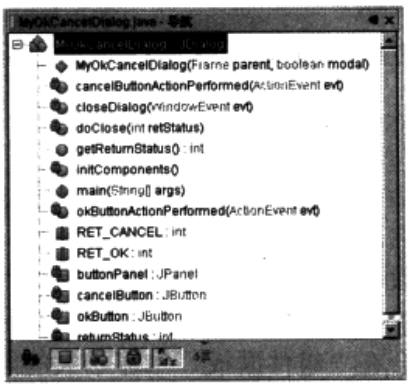


图 1-28 “导航”窗口

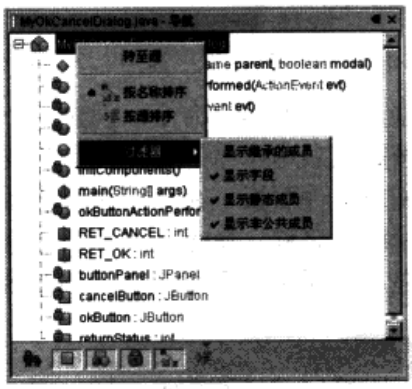


图 1-29 “导航”窗口的上下文菜单

“导航”窗口的上下文菜单如图 1-29 所示，包含以下命令。

- “转至源”命令：在源代码编辑器中打开类，并将插入点放在所选择方法的开始位置。
- “按名称排序”命令：按字母顺序显示成员。
- “按源排序”命令：按成员在代码中出现的顺序显示成员。
- “过滤器”命令：打开一个级联菜单，可以在其中为要显示的成员类型选择一些公共过滤器。可以选择以下过滤器的任意组合。
 - 显示继承的成员：显示来自当前类所扩展的类的成员。
 - 显示字段：显示类的字段。

Download at http://www.pin5i.com/

- 显示静态成员：显示静态字段和方法。
- 显示非公共成员：显示未提供公共访问权限的成员。



说明：默认情况下，将在集成开发环境的左下角打开“导航”窗口。也可以通过选择“窗口”→“输出”命令或者快捷键 Ctrl+7 打开“导航”窗口。要在“导航”窗口中显示源文件，请在源代码编辑器、“项目”窗口、“文件”窗口或“收藏夹”窗口中选择该文件。



1.4.6 “组件面板”窗口

“组件面板”窗口显示一个列表，其中包含可添加到集成开发环境的各种编辑器中的所有组件。对于 GUI 窗体，组件面板中的可用项包括容器和其他可视组件，如按钮、标签、菜单以及面板，如图 1-30 所示。

在“组件面板”中的任意空白位置单击鼠标右键，打开上下文菜单，如图 1-31 所示。在组件上单击鼠标右键，打开组件的上下文菜单，如图 1-32 所示。这两个弹出菜单的命令功能介绍如下。

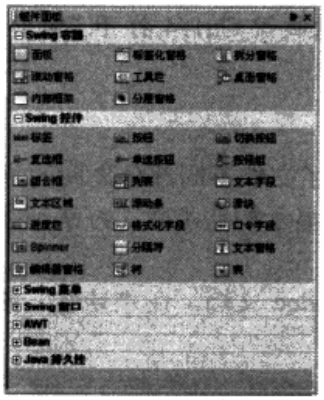


图 1-30 “组件面板”窗口



图 1-31 组件面板的上下文菜单



图 1-32 组件的上下文菜单

- “创建新的类别”命令：创建一个新的组件分类。
- “按类别名称排序”命令：按照类别的名字进行排序。
- “刷新组件面板”命令：在组件面板发生变化后，刷新组件面板，以显示最新的内容。
- “隐藏项名称”命令：选中该命令后，组件的名称将不显示（只显示图标）。
- “显示小图标”命令：选中该命令会在“组件面板”中显示小的组件图标。并且该命令会变为“显示大图标”命令。选中“显示大图标”命令，会在“组件面板”中显示大的组件图标，并切换回“显示小图标”命令。
- “重置组件面板”命令：把组件面板的设置恢复到默认的状态。
- “组件面板管理器...”命令：打开“组件面板管理器”对话框，如图 1-33 所示。
- “剪切”命令：剪切当前的组件。
- “复制”命令：复制当前的组件。
- “粘贴项”命令：把复制或者剪切的组件粘贴到制定的位置。

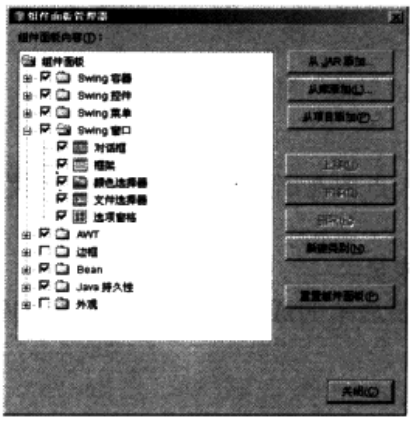


图 1-33 “组件面板管理器”对话框

使用“组件面板管理器”对话框，可以添加、删除

Download at http://www.pin5i.com/

和组织“组件面板”窗口中的组件，如 Swing 组件、AWT 组件、布局和 Bean。



说明：默认情况下，打开图形用户界面编辑器后，“组件面板”窗口也会自动打开。也可以通过选择“窗口”→“组件面板”命令或者快捷键 Ctrl+Shift+8 打开“组件面板”窗口。



1.4.7 “属性”窗口

在“项目”窗口中由节点表示的大多数对象及对象元素都具有属性，可以在“属性”窗口中查看/修改这些属性。“属性”窗口显示了“项目”窗口中当前选定节点的属性表单。

图 1-34 是 JButton 组件的属性表单，在这里可以查看和修改组件的属性。要更改属性值，请单击属性值字段。如果需要字符串值，则字段将变为文本字段。可以在此字段中直接键入新值，然后按回车键；如果允许使用特定的值列表，则会出现下拉箭头。可以单击此箭头，然后从出现的组合框中选择值。也可以双击属性名称以选择可用的值；如果属性编辑器适用于该属性，则会出现省略号（...）按钮。可以单击省略号按钮打开属性编辑器。

图 1-35 是 JButton 组件的绑定表单，当该组件和其他组件绑定后，在这里可以修改绑定源以及绑定表达式。



图 1-34 “属性”窗口的属性表单

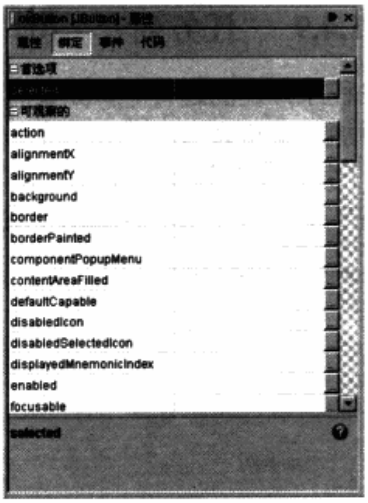


图 1-35 “属性”窗口的绑定表单

图 1-36 是 Jbutton 组件的事件表单，可以直接双击某个事件，然后在代码编辑器中为该事件添加代码。图 1-37 是 JButton 组件的代码表单，在这里可以修改组件的修饰符、变量名称等。



说明：默认情况下，打开图形用户界面编辑器后，“属性”窗口也会自动打开。也可以通过选择“窗口”→“属性”命令或者快捷键 Ctrl+Shift+7 打开“属性”窗口。



1.4.8 “检查器”窗口

“检查器”窗口显示当前打开的窗体中包含的所有组件的树状分层结构。显示的项包括可视组件和容器（如按钮、标签、菜单和面板）以及非可视组件（如计时器和数据源），如图 1-38 所示。

Download at http://www.pin5i.com/

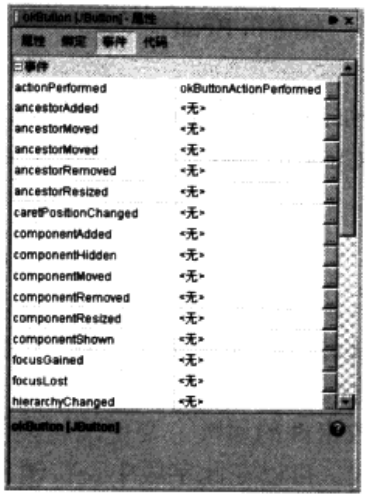


图 1-36 “属性”窗口的事件表单

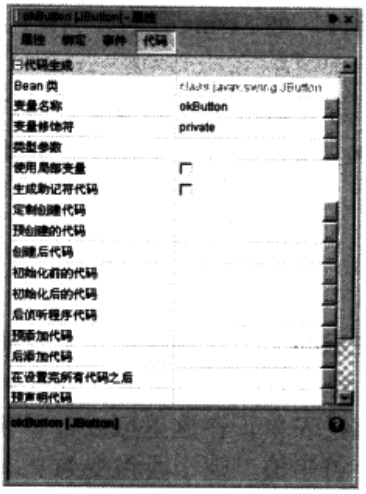


图 1-37 “属性”窗口的代码表单

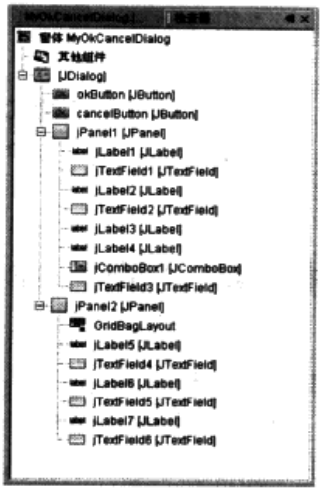


图 1-38 “检查器”窗口



说明：默认情况下，打开图形用户界面编辑器后，“检查器”窗口也会自动打开。也可以通过选择“窗口”→“检查器”命令打开“检查器”窗口。



1.4.9 “任务列表”窗口

在程序开发的过程中，如果有些事情需要以后再做，可以使用如图 1-39 所示的“//TODO”标记添加一个待做事项。“//TODO”的文字为待做事项的说明。

“任务列表”窗口中可以显示项目中待做任务的列表。对应如图 1-39 所示的待做任务，“待做任务”窗口的显示内容如图 1-40 所示。

```
private void doClose(int retStatus) {
    //TODO: 添加一个确认退出的对话框

    returnStatus = retStatus;
    setVisible(false);

    //XXX: 在退出前保存数据

    dispose();
}
```

图 1-39 添加待做事项

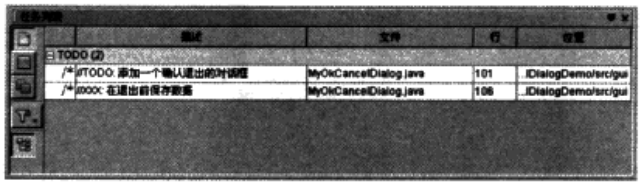


图 1-40 “任务列表”窗口

在“任务列表”窗口中执行以下操作，可以在源代码编辑器中跳转到一个待做任务：

- 双击某一个任务；
- 右键单击一个待做任务，然后选择弹出菜单的“显示源”命令。

通过单击任务列表的列标题，可以对任务列表进行排序。可以使用“任务列表”窗口的工具栏来修改和处理待做任务。

在“任务列表”窗口工具栏中，选定“当前文件”按钮，集成开发环境将扫描当前在源代码编辑器中打开的文件，并显示在该文件中找到的待做任务。在编辑器中打开新文件时，将扫描该文件，并且仅显示该文件中的待做任务。

在“待做任务”窗口工具栏中，如果选定“打开的项目”按钮，集成开发环境将扫描所有打开的项目，并显示这些项目中的所有待做任务。如果选定“主项目”按钮，集成开发环境将扫描当前主项目中的所有待做任务。

在“待做任务”窗口工具栏中，单击“编辑过滤器”按钮，打开“任务列表过滤器”

Download at http://www.pin5i.com/

对话框，如图 1-41 所示。在该对话框中可以添加和修改过滤器，以过滤“任务列表”窗口中显示的待做任务。

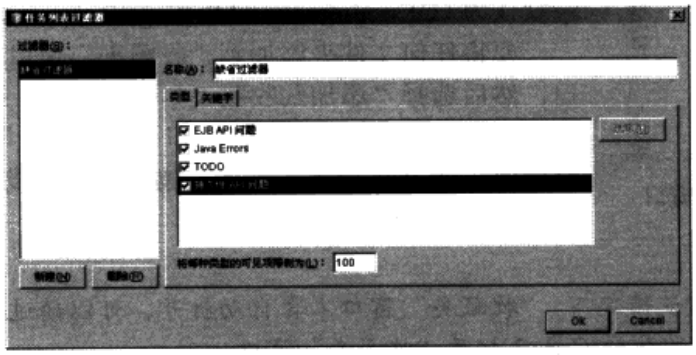


图 1-41 “任务列表过滤器”对话框

下面几种方法都会创建一个待做任务，并自动被添加到“任务列表”窗口中：

- 包含 TODO、XXX 或 FIXME 的代码注释；
- 模式<<<<<<<，它在 CVS 和其他版本控制系统中表示合并冲突；
- Java 项目中，包含@todo 关键字的 Javadoc 注释；
- Java 项目中，登记编译器错误时所在的行。

NetBeans 集成开发环境支持添加或者编辑待做任务标识。单击“工具”→“选项”命令，打开“选项”对话框。单击“杂项”按钮，再打开“待做任务”标签，如图 1-42 所示。在这里，用户可以根据自己的习惯设置待做任务的标识，以这些标识开头的内容将会自动添加到“任务列表”窗口中。

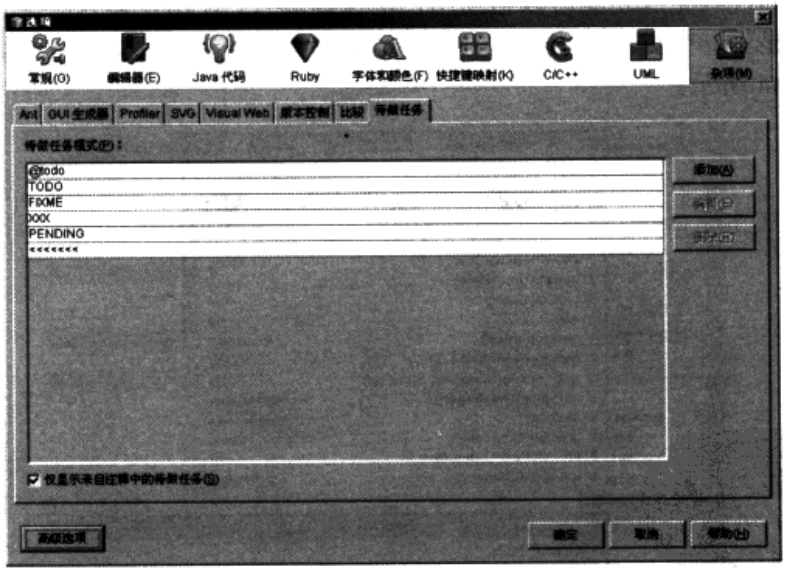


图 1-42 “待做任务”标签

说明：默认情况下，“任务列表”窗口不会自动打开，可以通过选择“窗口”→“任务列表”命令或者快捷键 Ctrl+6 打开“任务列表”窗口。

1.4.10 “收藏夹”窗口

使用“收藏夹”窗口，开发人员可以访问计算机或网络上的任何文件，而无须为该文件创

Download at http://www.pin5i.com/

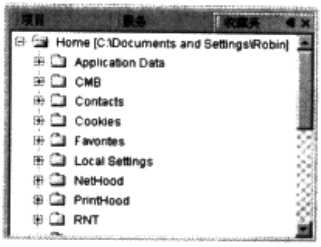


图 1-43 “收藏夹”窗口

建项目。首次打开“收藏夹”窗口时，它只包含计算机的 Home 目录。对于 Windows 系统，如果登录用户为 Robin，那么这个目录默认为 C:\Documents and Settings\Robin，如图 1-43 所示。

要将任何文件夹添加到“收藏夹”窗口中，请右键单击该窗口，然后选择“添加到收藏夹”。“收藏夹”窗口中不提供任何常规项目命令（如“运行文件”和“编译文件”）。可以从“收藏夹”窗口中运行 Ant 脚本，就像在集成开发环境中的任何其他地方运行 Ant 脚本一样。

说明：默认情况下，“收藏夹”窗口不会自动打开，可以通过选择“窗口”→“收藏夹”命令或者快捷键 Ctrl+3 打开“收藏夹”窗口。

1.5 源代码编辑器

NetBeans 集成开发环境中，使用最多的就是源代码编辑器。它提供了各种可以使编写代码更简单、更快捷的功能，如代码完成、突出显示编译错误、代码重构、代码搜索等功能。本节将介绍 NetBeans 源代码编辑器的几个功能。

1.5.1 快速编写代码

在 NetBeans 中输入代码时不必手工输入很长的 Java 类名、表达式或者方法名，NetBeans 代码编辑器提供的代码完成功能可以帮助用户快速地输入。在用户输入了字符后，NetBeans 代码编辑器就可以显示一个提示菜单，给出包含可能的类、方法和变量等的列表，以便于用户选择输入，如图 1-44 所示。

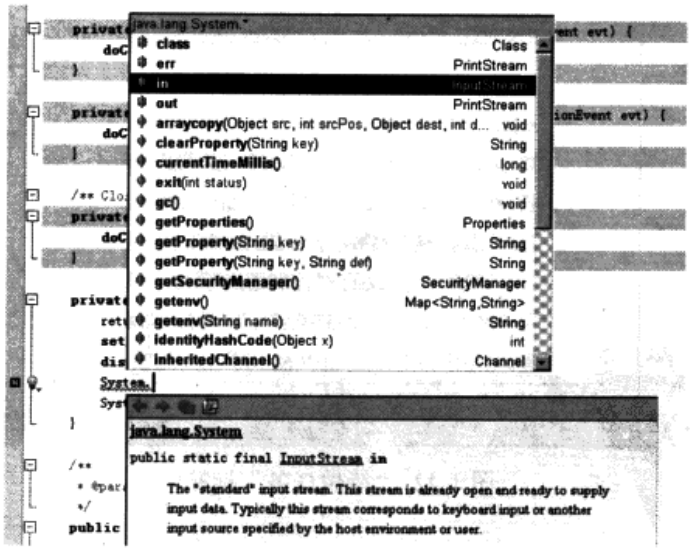


图 1-44 代码自动完成功能

另外，用户输入某些字符后，按下 Ctrl + SPACE（空格）组合键或者 Ctrl+\组合键，也会显示与输入的字符相关的提示信息以及 Javadoc 文档。例如，用户输入“for”后，按下 Ctrl + SPACE（空格）键，就会显示如图 1-1 所示提示信息。

Download at http://www.pin5i.com/

注意：这里推荐读者使用快捷键 Ctrl+\组合键，因为中文 Windows 系统中 Ctrl + SPACE 能够打开和关闭输入法，和 NetBeans 中的快捷键冲突。如果一定要使用 Ctrl + SPACE 键，那么需要把输入法开关键修改为其他按键。

1.5.2 代码模板

NetBeans 的源代码编辑器支持代码模板功能（在以前的 NetBeans 版本中被称为“缩写”），为键入常用的击键组合和字符串提供了快捷方式。在源代码编辑器中，键入代码模板的缩写，然后按 Tab 或空格键以生成完整的代码片段。例如，键入 psfs，再按 Tab 键，就在源代码编辑器中自动生成 public static final String。

单击“工具”→“选项”命令，打开“选项”对话框。单击“编辑器”按钮，再打开“代码模板”标签，就可以查看已经定义的代码缩写，如图 1-45 所示。

单击“新建”按钮，打开“新建代码模板”对话框，如图 1-46 所示。例如希望添加 try...catch 代码块的模板，这里就可以输入 tc。单击“确定”按钮，返回“选项”对话框，在“展开的文本”中输入如下的内容：

```
try {
    ${cursor}
} catch (Exception e) {
    e.printStackTrace();
}
```

其中，${cursor}表示的是光标的位置。单击“确定”按钮，返回源代码编辑器，这时就可以使用刚刚创建的代码模板了。在源代码编辑器中输入 tc，然后按 Tab 或空格键，即可自动添加上面编辑的代码块了。

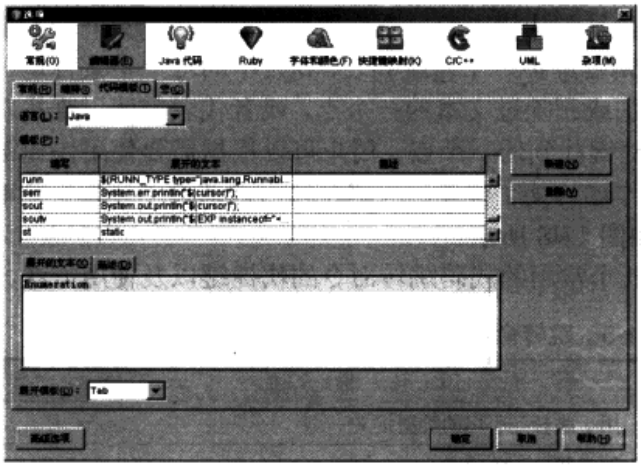


图 1-45 “代码模板”标签

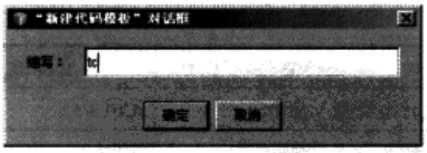


图 1-46 “新建代码模板”对话框

注意：在 NetBeans 6.0 以前的版本中，都是使用空格键来展开代码缩写的。在 NetBeans 6.0 中替换为使用 Tab 键了。如果读者仍然希望使用空格键，可以在图 1-45 所示的界面中打开“展开模板”下拉列表框，然后选择“空格”选项。

1.5.3 格式化 Java 代码

在编写代码时一般都需要遵循一定的编码规范，包括统一的命名格式、统一的缩进格式。在

Download at http://www.pin5i.com/

源代码编辑器中键入一行代码并按回车键时，新行会根据代码层次和所选的缩排引擎自动缩排。

如果代码中的缩进格式不规范，进行手工调整会非常麻烦。NetBeans 的代码编辑器可以帮助用户快速地格式化代码。下面的几种方式都可以实现代码重新格式化：

- 使用快捷键 Alt + Shift + F 组合键（在 NetBeans 6.0 以前的版本中使用的是 Ctrl + Shift + F 组合键）；
- 在源代码编辑器中单击右键，选择弹出菜单中的“格式”命令；
- 选择菜单“源”→“格式”命令。

图 1-47 中，左边的图为格式化前的代码，右边的图为格式化后的代码。默认情况下，重新格式化操作针对的是整个源文件。如果只希望格式化部分代码，那么可以在源代码编辑器中选中要格式化的代码，然后执行格式化操作。

```
private void doClose(int retStatus) {
                    returnStatus = retStatus;
    setVisible(false);
                dispose();
        System.exit(0);

}
```

```
private void doClose(int retStatus) {
    returnStatus = retStatus;
    setVisible(false);
    dispose();
    System.exit(0);

}
```

图 1-47　重新设置代码格式



图 1-48　“导航”菜单

技巧：NetBeans 还支持将代码粘贴到文件中时自动设置代码格式。在复制要放到源代码编辑器中的文本后，在源代码编辑器中，单击要放置代码的位置，然后按 Ctrl+Shift+V 组合键即可。

1.5.4　代码跳转

所谓代码跳转就是指从代码的一个位置快速移动到另外一个位置。NetBeans 源代码编辑器维护一个转跳列表，列表中的每一个条目均代表文件中用户曾经添加或删除过文本的一个点，或者代表曾使用查找功能或“转至”导航快捷键找到的一个点。用户可以根据需要使用快捷键跳到文件中您先前工作过的某个位置。代码跳转可以通过“导航”菜单实现，“导航”菜单如图 1-48 所示。

表 1-5 给出了几个常用的代码跳转命令的快捷键以及使用方法。

表 1-5　跳转命令

跳转命令	快捷键	描　述
“转至类型”命令	Ctrl + O	用于跳转到引用的类型定义
“转至测试”命令	Alt + Shift + T	用于跳转到测试类
“至上一文档”命令	Ctrl +反引号	到上一个刚刚访问过的文档
“转至源文件”命令	Alt + Shift + B	转至插入点所在的标识符的 Java 源代码
“转至声明”命令	Ctrl + B	跳到插入点所在的 Java 方法或字段的声明部分
“上一个编辑位置”命令	Ctrl + Q	跳转到上一次编辑的位置
“后退”命令	Alt + Left	移到转跳列表中的上一个条目，这类似于在浏览器中使用“后退”按钮，该命令与“前进”命令相对应
“前进”命令	Alt + Right	移到转跳列表中的下一个条目，这类似于在浏览器中使用“前进”按钮；该命令和“后退”命令相对应，只有在已经返回后才可用

Download at http://www.pin5i.com/

续表

跳转命令	快捷键	描　述
“转至行”命令	Ctrl + G	转至特定行号
“切换书签”	Ctrl + Shift + M	在代码行上添加书签，在包含书签的行上按 Ctrl + F2 组合键可删除书签
“下一个书签”命令	Ctrl + Shift + 句点	移到下一个书签所在的行
“上一个书签”命令	Ctrl + Shift +逗号	移到上一个书签所在的行
“下一个错误”命令	Ctrl +句点	移到下一个错误所在的行
“上一个错误”命令	Ctrl +逗号	移到上一个错误所在的行



技巧：按下 Ctrl 键，同时将鼠标移到某个标识符并单击它也可以实现跳转。



1.6 更新 NetBeans

NetBeans 还提供了更新功能，用户可以安装更多的 NetBeans 模块以扩展 NetBeans 集成开发环境的功能。更新 NetBeans 有以下两种方法。

- 自动联网更新：直接从互联网上进行更新。其特点是简单，不需要过多的操作即可完成更新。
- 手工更新：进行手工更新时，首先要下载需要更新的插件模块（*.nbm）文件，然后再手工安装到 NetBeans 集成开发环境中。手工更新的特点就是不依赖于网络，但是操作相对烦琐，需要知道在哪里下载插件模块文件，并需要手工安装。

自动联网更新将在第 2 章的第 2.1.3 节进行详细的介绍，下面主要介绍如何进行手工更新。这里，假定已经下载了插件模块文件。

单击“工具”→“插件”命令，打开“插件”对话框，打开“已下载”标签页，如图 1-49 所示。在这个对话框中列出了所有手工安装的插件模块。

单击“添加插件”按钮，打开“添加插件”对话框。通过该对话框选择下载的.nbm 文件，并完成安装。安装完毕后，“已下载”标签页会如图 1-50 所示。

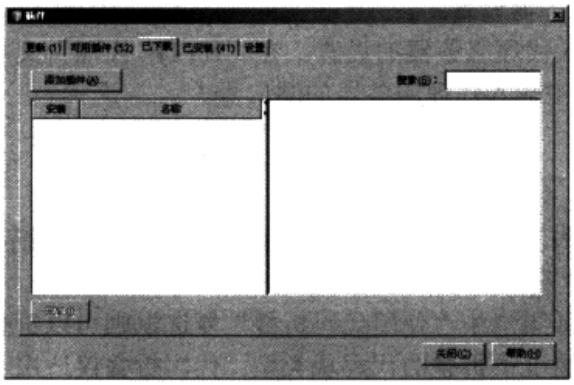


图 1-49 “插件”对话框的“已下载”标签页

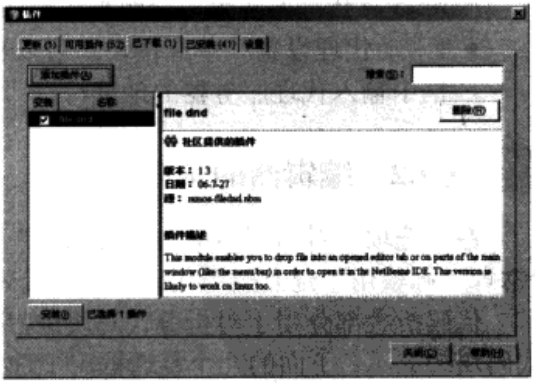


图 1-50 手工安装插件后的“已下载”标签页



1.7 配置 NetBeans 开发环境

集成开发环境允许开发人员为其许多功能配置选项，还允许定制工作区。下面介绍 NetBeans 配置选项的功能和使用方法。

Download at http://www.pin5i.com/

1.7.1 常规配置

选择“工具”→“选项”命令，打开“选项”对话框，单击左边的“常规”按钮可以进行一些常规的设置，如图 1-51 所示。

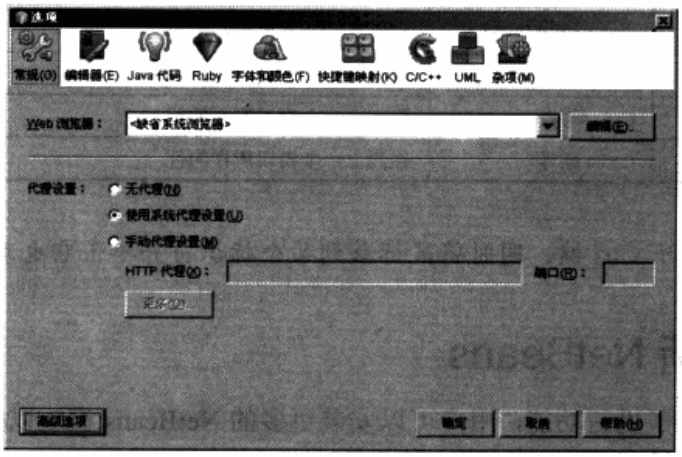


图 1-51 常规配置

“Web 浏览器”下拉列表框用于设置 NetBeans 默认的 Web 浏览器。

代理部分用于设置上网的代理服务器。在“代理配置”对话框中有三个单选按钮，其含义分别介绍如下。

- “无代理”单选按钮：不使用代理连接互联网。
- 使用系统代理设置”单选按钮：使用系统的代理服务器设置。如果要修改系统的代理设置，可以按照如下的步骤操作。
 - 选择 IE 浏览器的“Internet 选项”命令，或者双击控制面板中的“Internet 选项”图标，打开“Internet 选项”对话框。
 - 选择“连接”选项卡。
 - 单击“局域网设置”按钮，打开“局域网（LAN）设置”对话框。
 - 在“代理服务器”分组框中修改代理服务器设置。
- “手动代理设置”单选按钮：可以自己输入代理服务器的设置。在“HTTP 代理”文本框中输入代理服务器的地址，在“端口”中输入代理服务器的端口号。

1.7.2 编辑器的常规选项

在“选项”对话框中单击“编辑器”按钮，打开编辑器的“常规”标签，如图 1-52 所示。“常规”标签用于定制全局代码折叠和代码完成设置。

代码折叠部分用于对折叠代码块进行控制。如果选中“使用代码折叠”复选框则可以折叠代码块的显示。下面的几个复选框用于选择默认情况下折叠的代码块类型。代码折叠仅适用于支持代码折叠的编辑器类型，如 Java 编辑器。

在源代码编辑器中键入内容时，可以使用集成开发环境的代码完成功能来查找并自动填写代码片段。编辑器常规选项的代码完成部分，可以对代码完成设置进行控制。

- “自动弹出式完成窗口”复选框：如果选定该属性，编辑器可以显示一个代码完成框，其中包含如何完成表达式的选项。此属性只适用于支持代码完成的编辑器类型，如 Java、HTML 和 JSP。

Download at http://www.pin5i.com/

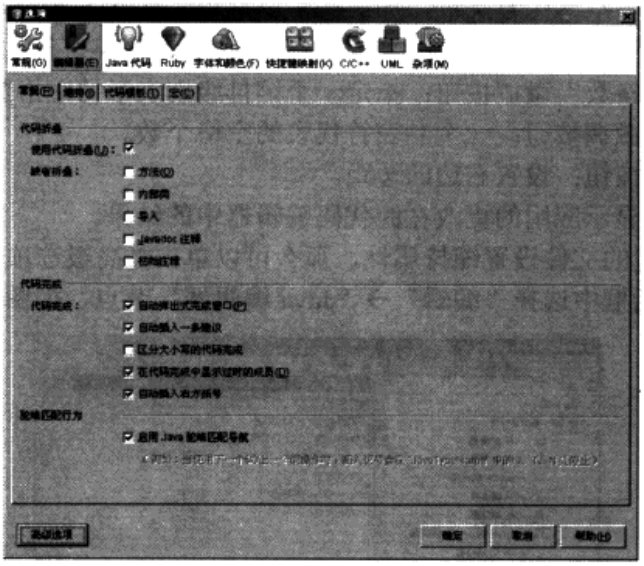


图 1-52 编辑器的常规选项

- “自动插入一条建议”复选框：如果选定该属性，当集成开发环境知道表达式只有一种可能的代码完成情况时，将不显示代码完成框，自动完成该表达式。
- “区分大小写的代码完成”复选框：如果选定该属性，则代码完成框将仅显示与要键入的表达式大小写相匹配的结果。
- “在代码完成中显示过时的成员”复选框：如果选定该属性，完成框中将显示任何可用的过时类元素。过时成员带有删除线以表明它们已过时。
- “自动插入右方括号”复选框：如果选定该属性，在键入左圆括号、方括号、花括号或引号后，源代码编辑器将生成相应的右括号或引号。



1.7.3 编辑器的缩排选项

在“选项”对话框中单击“编辑器”按钮，再选择“缩排”标签，如图 1-53 所示。在这里可以设置代码的缩排格式。

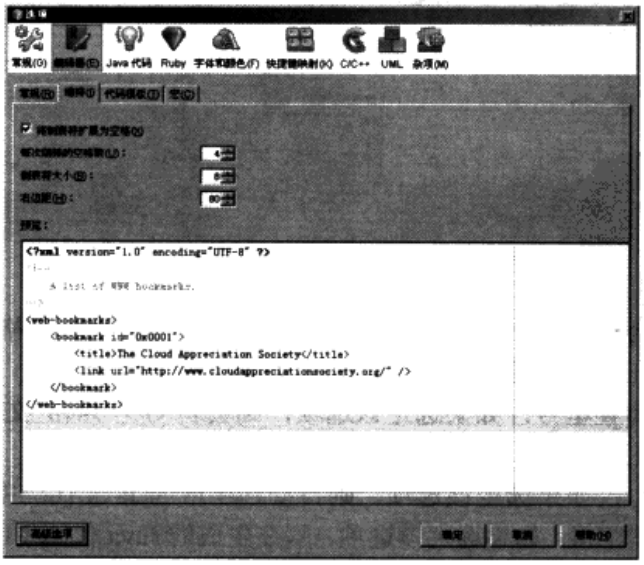


图 1-53 设置代码的缩排属性

Download at http://www.pin5i.com/

- “将制表符扩展为空格”复选框：如果选定该选项，按 Tab 键时将插入空格。如果未选定该选项，按 Tab 键时将在文件中插入制表符（\t）（如果可能的话）。
- “每次缩排的空格数”微调按钮：指定一个缩排级别表示的空格数。
- “制表符大小”微调按钮：一个制表符代表的空格个数。
- “右边距”微调按钮：设置右边的边距。
- “预览”窗格：显示应用的更改在源代码编辑器中的结果。

如果要为其他类型的文件设置缩排属性，那么可以单击“高级选项”按钮，打开“高级选项”对话框。在该对话框中选择“编辑”→“Java 编辑器”节点，如图 1-54 所示。

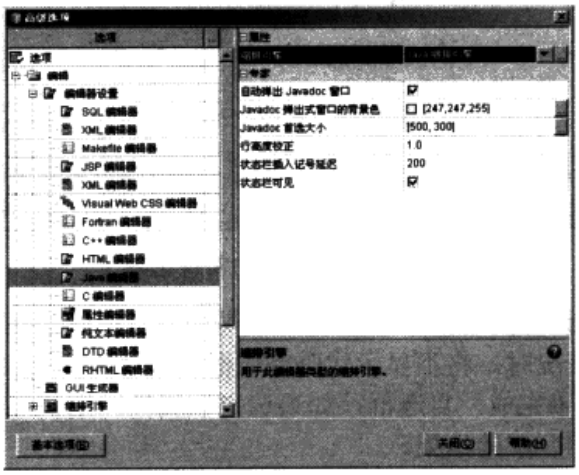


图 1-54 “高级选项”对话框

此时，可以在右边的“缩排引擎”字段中通过下拉列表框选择不同的缩排引擎，也可以单击...按钮打开“Java 编辑器–缩排引擎”对话框，如图 1-55 所示。其中各个选项的含义介绍如下。

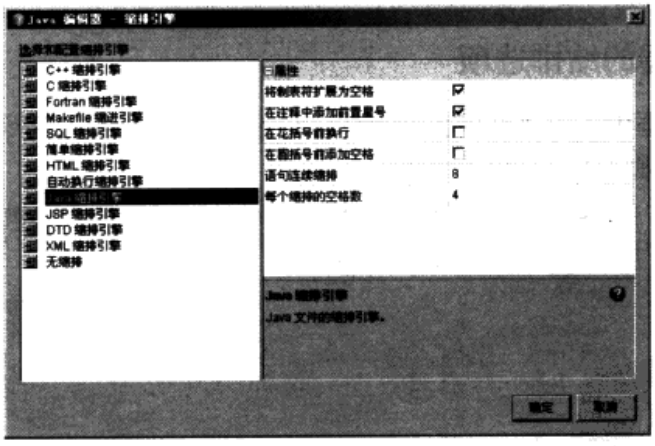


图 1-55 “Java 编辑器–缩排引擎”对话框

- 将制表符扩展为空格：如果选定该选项，按 Tab 键时将插入空格。如果未选定该选项，按 Tab 键时将在文件中插入制表符（\t）（如果可能的话）。
- 在注释中添加前置星号：如果选定该选项，则会自动在 Java 文件的块注释内的所有行开头添加一个星号（*）。
- 在花括号前换行：如果选定该选项，则自动在生成的 Java 代码中将花括号移到下一行。
- 在圆括号前添加空格：如果选定该选项，则在生成的 Java 代码中的左圆括号前添加空格。
- 语句连续缩排：为多行语句中的第二行及后续行指定缩排的空格数。

Download at http://www.pin5i.com/

- 每个缩排的空格数：指定一个缩排级别表示的空格数。

1.7.4 定制代码模板

在“选项”对话框中单击“编辑器”按钮，再选择“代码模板”标签，如图 1-56 所示。NetBeans 选项对话框中的“代码模板”标签用于定制希望编辑器自动展开的词语缩写。

“语言”下拉列表框用来选择编辑的是哪个语言的代码模板。“新建”按钮可以添加一个新的代码模板。“删除”按钮可以删除“模板”列表中选中的模板。“展开的文本”文本框用于输入/修改当前模板展开后的实际内容。“展开模板”下拉列表框可以更改用于展开缩写的键或组合键（默认情况下，按 Tab 键时将展开缩写）：

- 空格（Space）
- Shift+空格键
- 回车键（Enter）
- 跳格键（Tab）

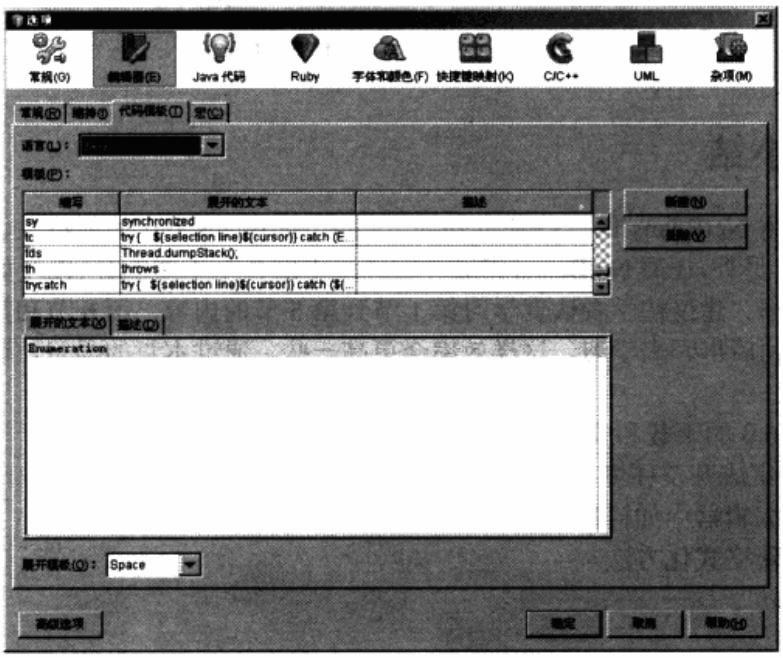


图 1-56 定制代码模板

1.7.5 字体和颜色

在“选项”对话框中单击“字体和颜色”按钮，打开“字体和颜色”面板，如图 1-57 所示。此面板用于为源代码编辑器中的文本和元素定制字体、背景色和前景色。

可以创建配置文件来保存各种定制字体和颜色设置，也通过配置文件 Profile 下拉列表框在配置文件之间快速切换。如果修改了集成开发环境中包含的某个配置文件，则可以单击“恢复”按钮将选定的配置文件恢复为默认设置。单击“复制”按钮则会复制当前配置文件的设置。

“语法”标签可以根据类型和语言来定制文本外观，例如可以指定字体类型、字体颜色和背景色等。

“突出显示”标签用来定制突出显示区域的前景色和背景色，其中包括代码折叠以及突出显示元素的显示方式。“标注”标签可以定制图标旁注中标注图标的前景色和背景色。

Download at http://www.pin5i.com/

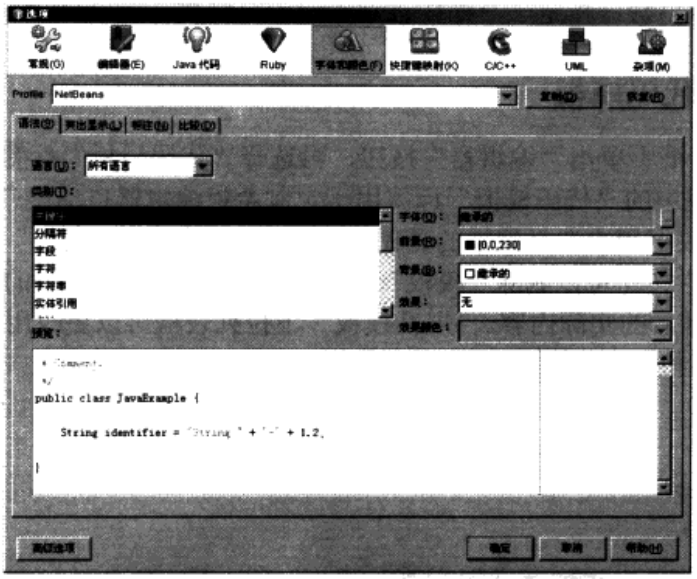


图 1-57 定制字体和颜色

1.8 小结

本章主要介绍 NetBeans 的一些基本知识，包括 NetBeans 6.0 的新特性、如何下载和安装 NetBeans 等内容。另外，还对 NetBeans 集成开发环境中的各个主要部分、以及如何配置 NetBeans 进行了详细的介绍。建议初学者认真学习第 1 节到第 5 节的内容，第 6 节和第 7 节进行快速浏览即可。以后用到后再回来学习，这样效率会更高一些。通过本章的学习，读者应该掌握以下的内容：

- NetBeans 6.0 的下载和安装方法
- NetBeans 集成开发环境中各个窗口的使用方法
- 在源代码编辑器中如何快速输入代码
- 代码的重新格式化方法
- 代码的跳转

Download at http://www.pin5i.com/

第2章

通过 NetBeans 开发 Java 标准应用程序

通过 NetBeans 集成开发环境可以快速地创建 Java 标准应用程序，利用 NetBeans 提供的可视化编程界面可以为应用程序添加/编辑界面组件、编译运行项目并最后发布项目。本章将讨论如何在 NetBeans 集成开发环境中完成这些基本的任务，快速开发 Java 标准应用程序。本章的讲解将以一个实际项目的开发过程为基础，一步一步地介绍 Java 标准项目在 NetBeans 集成开发环境中的开发实现。

本章主要内容

- 如何通过模板创建标准的 Java 应用程序
- 如何添加和编辑界面组件
- 项目的编译运行
- 如何配置和管理项目
- 如何使应用程序支持国际语言
- 如何打包和发布应用程序

本章亮点

- 在 NetBeans 集成开发环境中添加并编辑界面组件
- NetBeans 对国际化开发的支持
- NetBeans 项目的管理

2.1 创建 Java 标准应用程序

在 NetBeans IDE 中，开发工作都将在项目内完成，NetBeans IDE 中的项目由一组源文件以及用来生成、运行和调试这些源文件的设置组成。这些设置包括哪些内容适合放在类路径中、如何生成和运行项目等。

2.1.1 创建 Java 标准应用程序项目

使用 NetBeans 集成开发环境生成的 Ant 脚本来编译、运行和调试应用程序的项目被成为标准项目。通过 NetBeans 集成开发环境生成的大部分项目都是标准项目，下面介绍生成标准项目的具体方法。

❶ 在 NetBeans 集成开发环境中，选择“文件”→“新建项目”命令，打开“新建项目”对话框，如图 2-1 所示。

在“类别”列表框中选择“Java”节点，此时会在右边的“项目”列表框中显示如下几个节点。

- Java 应用程序：通过“新建 Java 应用程序”向导，可以创建 Java 标准应用程序的项目。
- Java 桌面应用程序：创建一个基于 Swing 组件的桌面应用程序框架。此模板提供基本的

Download at http://www.pin5i.com/

应用程序基础结构，如菜单栏、永久保持窗口状态以及状态栏。利用此模板，还可以生成代码以便为数据库表创建 GUI 界面。

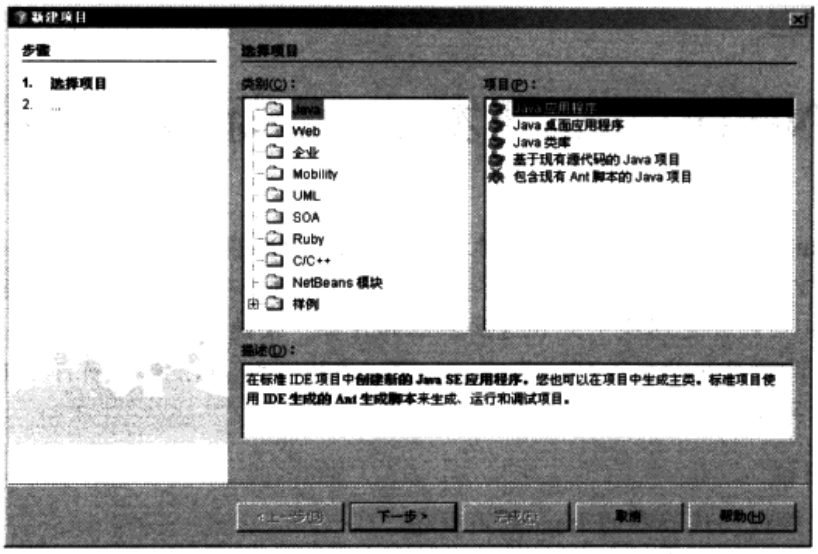


图 2-1 “新建项目”对话框

- Java 类库：通过“新建 Java 类库”向导，可以创建不包含主类的新标准项目，该项目主要用来创建 Java 标准类库。
- 基于现有源代码的 Java 项目：通过“新建包含现有源的 Java 项目”向导，可以使用现有的 Java SE 源文件夹创建新的标准项目。
- 包含现有 Ant 脚本的 Java 项目：将现有的 Java 模块导入到自由格式项目中。

❷ 选中“Java 应用程序”节点，单击“下一步”按钮，打开“名称和位置”对话框，如图 2-2 所示。

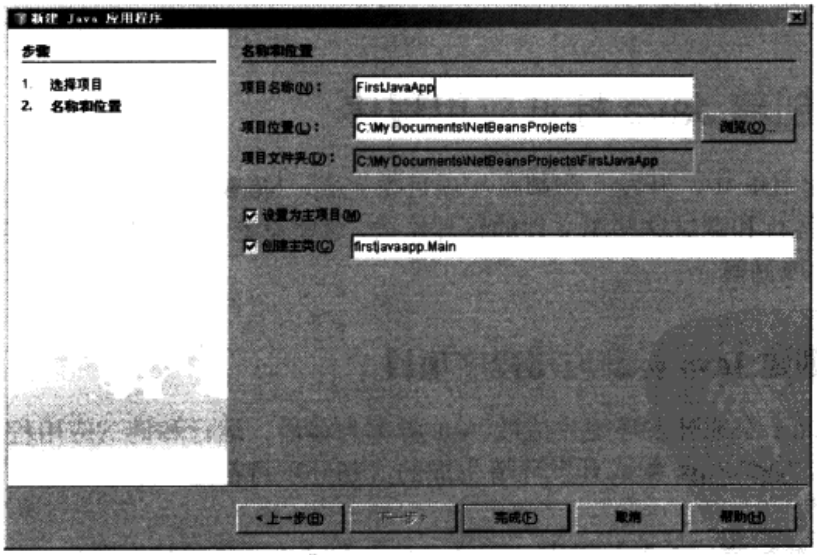


图 2-2 “名称和位置”对话框

❸ 在“项目位置”文本框中输入项目的目录，或者单击“浏览”按钮选择项目的目录。在“项目名称”文本框中输入项目的名字。如果希望把创建的项目作为主项目，那么需要选中“设置为主项目”复选框。

Download at http://www.pin5i.com/

名词解释：主项目

在打开多个项目时，应始终将其中的一个项目设置为主项目。主项目通常是包含应用程序主类的项目。IDE 提供了可快速生成、运行和调试主项目的快捷命令。在“项目”窗口中，主项目以粗体显示。可以将任何项目设置为主项目，方法是：右键单击该项目的项目根节点，然后选择“设置为主项目”命令即可。

❹ 如果希望同时创建主类，可以选中“创建主类”复选框，并在后面的文本框中输入主类的路径和名称。

❺ 单击“完成”按钮，完成 Java 标准项目的创建，至此创建了一个简单的 Java 标准应用程序。创建的 Java 标准项目在“项目”窗口中的显示如图 2-3 所示，包含源包、测试包、库、测试库等几个部分。

创建的项目文件夹如图 2-4 所示。NetBeans IDE 将项目信息存储在项目文件夹的根文件夹和 nbproject 文件夹中，包括 Ant 生成脚本、控制生成和运行设置的属性文件，以及若干个 XML 配置文件（例如，将 Ant 目标映射到 NetBeans IDE 命令的 project.xml 文件）。源目录一般都包含在项目文件夹中，目录名字为 src，但这并不是必需的。test 目录通常用于保存项目的测试包。



图 2-3 “项目”窗口中的 Java 标准项目

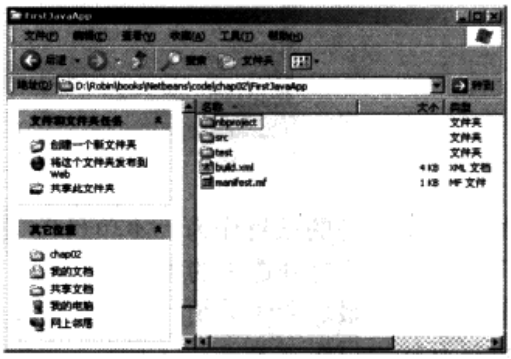


图 2-4 Java 标准项目文件夹

在源代码编辑器中打开 Main.java 文件，可以看到 NetBeans 自动生成的代码如下所示：

```
/*
 * Main.java
 *
 * Created on 2006年10月15日, 下午5:36
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package firstjavaapp;

/**
 *
 * @author Robin
 */
public class Main {

    /** Creates a new instance of Main */
    public Main() {
    }

    /**
```

Download at http://www.pin5i.com/


```
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```



2.1.2 打开和关闭项目

1. 打开项目

在 Netbeans 中可以根据需要关闭某个项目，或者打开一个已经存在的项目。打开项目通过选择“文件”→“打开项目”命令（快捷键为 Ctrl+Shift+O），或者单击工具栏上的“打开项目”图标 。在弹出的“打开项目”对话框中，选择项目的根文件夹，如图 2-5 所示。

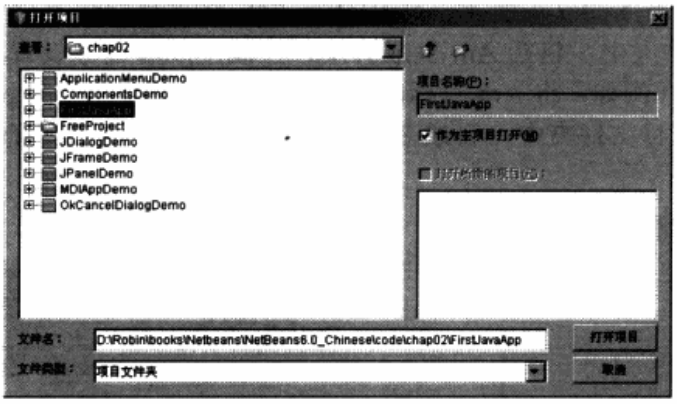


图 2-5 “打开项目”对话框

如果希望打开的项目作为“主项目”，就要选中“作为主项目打开”复选框。单击“打开项目”命令，即可打开选中的项目。

如果要打开最近刚刚关闭的项目，可以通过选择“文件”→“打开最近的项目”命令，打开最近打开的项目级联菜单，里面会列出最近的 4 个项目，如图 2-6 所示。在级联菜单中选择某个项目，即可再次打开该项目。

2. 关闭项目

如果要关闭某个项目，可以通过下面的两种方法进行。

❶ 在“项目”窗口中选中要被关闭项目的根节点或者其子节点（即选中必须在被关闭的项目范围内），然后选择“文件”→“关闭 xxx”命令。其中“xxx”为被选中项目的名称，根据项目的不同，其显示内容也不相同。

❷ 在“项目”窗口中该项目根节点上单击右键，然后选择“关闭”命令，如图 2-7 所示。



2.1.3 导入项目

NetBeans IDE 的一个优秀特性就是不仅支持打开/关闭 NetBeans 项目，还支持打开 Eclipse 和 JBuilder 开发的项目。这样，开发人员可以在最短的时间内，把 Eclipse 或者 JBuilder 的项目移植到 NetBeans 集成开发环境中。

1. 安装导入模块

要导入 Eclipse 项目，必须首先安装 Eclipse 项目导入模块；要导入 JBuilder 项目，必须首先安装 Eclipse 项目导入模块。下面具体介绍这两个模块的安装步骤。

Download at http://www.pin5i.com/

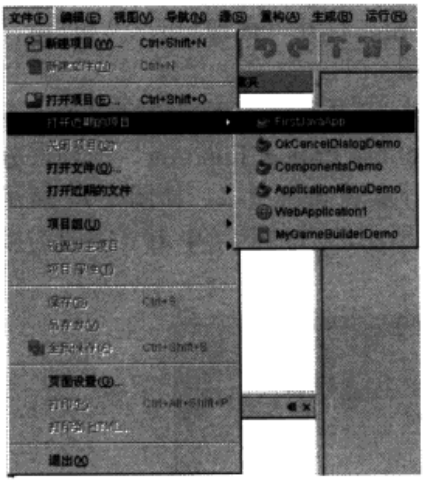


图 2-6 最近打开的项目级联菜单

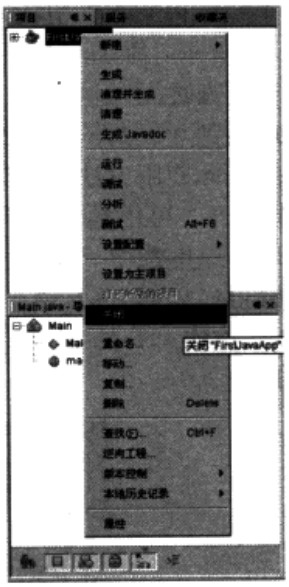


图 2-7 关闭项目

❶ 选择“工具”→“插件”命令，打开“插件”对话框。NetBeans 提供了两种方式安装模块：一种是通过联网在线安装；另外一种是在本地安装已经下载的模块文件（*.nbm）。在“已下载”标签页中提供了安装手工下载的模块（*.nbm 文件）的功能。这里选择在线安装，因此要首先设置网络连接。打开“设置”标签页，如图 2-8 所示。

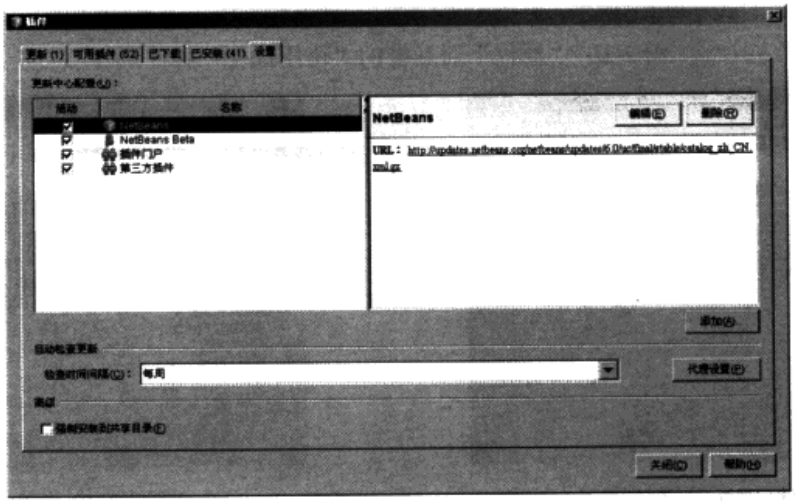


图 2-8 “插件”对话框的“设置”标签页

❷ 在标签页左边的列表中给出了所有可用的更新中心。单击“添加”按钮，打开“更新中心定制器”对话框，可以添加其他的更新中心，如图 2-9 所示。

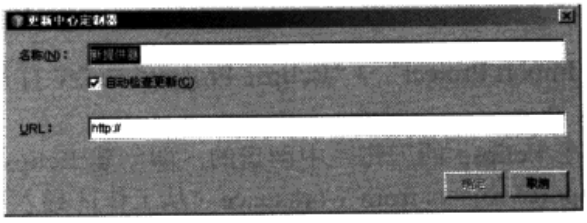


图 2-9 “更新中心定制器”对话框

Download at http://www.pin5i.com/

在“名称”文本框中输入更新中心的名字，在 URL 文本框中输入更新中心的地址，单击“确定”按钮即可添加一个新的更新中心。

在“检查时间间隔”下拉列表框中可以选择检查更新的时间间隔。如果计算机是通过代理服务器连接到互联网的，那么需要单击“代理设置”按钮，打开“代理设置”对话框，关于该对话框的具体使用方法，请读者参考第 1.7.1 节中的介绍。

❸ 打开“可用插件”标签页，如图 2-10 所示。在标签页左边的列表中列出了当前所有可用的模块，这里选中“JBuilder Project Importer”和“Eclipse Project Importer”选项，分别用于导入 JBuilder 和 Eclipse 项目。单击“安装”按钮，打开安装确认窗口，如图 2-11 所示。单击“下一步”按钮，开始安装，安装完毕后显示如图 2-12 所示的对话框。单击“完成”按钮，完成安装过程。

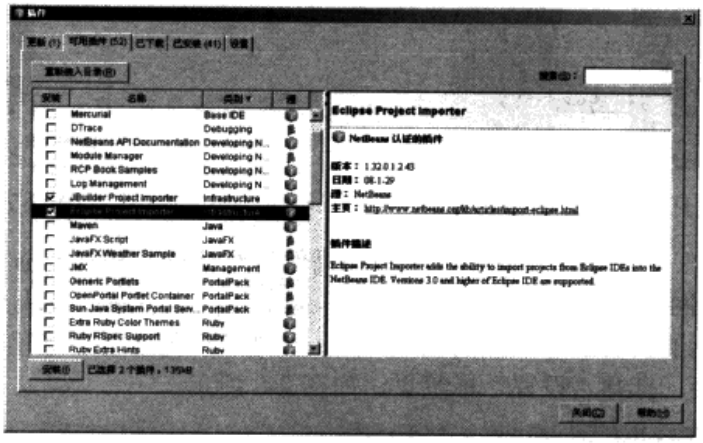


图 2-10 “插件”对话框的“可用插件”标签页

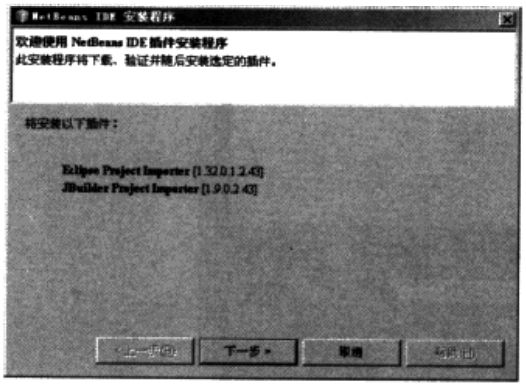


图 2-11 安装确认窗口

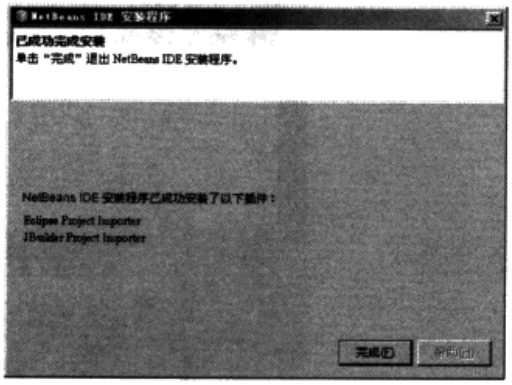


图 2-12 完成安装

2. 导入 Eclipse 和 JBuilder 项目文件

在安装了导入模块后，会在“文件”菜单下添加一个“Import Project”级联菜单，如图 2-13 所示。通过这个级联菜单，就可以导入相应的项目了。下面以 Eclipse 项目为例介绍导入项目的具体操作步骤。

❶ 选择“文件”→“Import Project”→“Eclipse Project”命令，打开“Import Eclipse Project”对话框，如图 2-14 所示。

如果 Eclipse 项目是在 Eclipse 的工作区中创建的（即创建 Eclipse 项目时选择的是默认的目录），那么需要选中 Import Projects from Workspace（从工作区导入项目）单选按钮，然后在 Workspace Location（工作区位置）文本框中输入 Eclipse 工作区的位置。输入的不是 Eclipse 的工作区目录，否则会显示如图 2-15 所示的错误。

Download at http://www.pin5i.com/

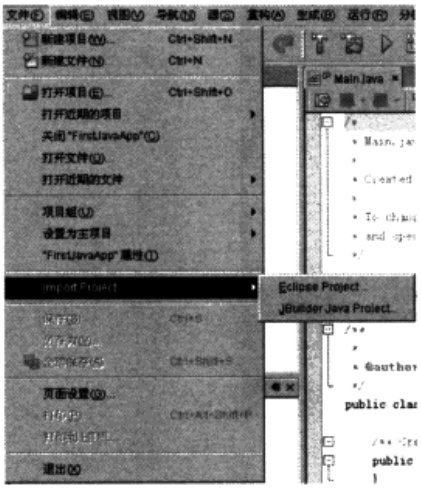


图 2-13 “Import Project”级联菜单

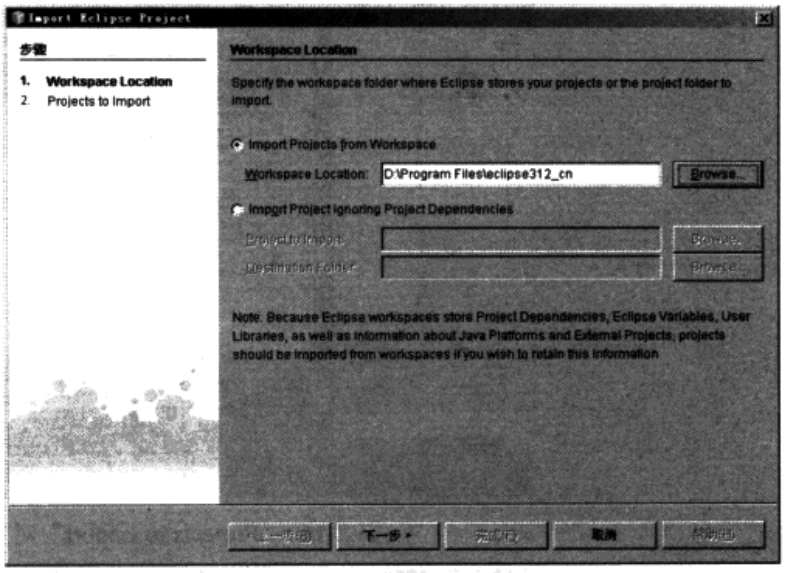


图 2-14 “Import Eclipse Project”对话框

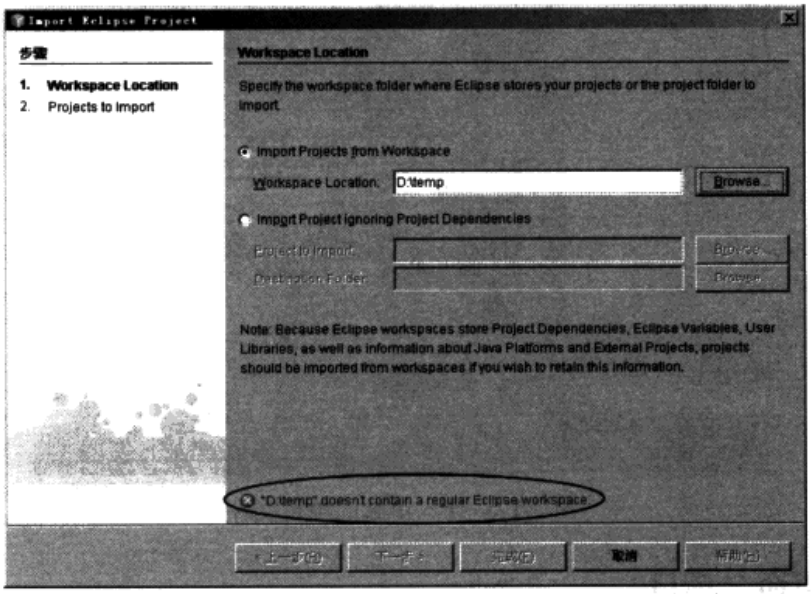


图 2-15 工作区位置错误时的提示信息

❷ 在图 2-14 中，单击“下一步”按钮，打开“Projects to Import”对话框，如图 2-16 所示。Projects to Import（要导入的项目）列表中列出了 Eclipse 工作区中的全部项目。选中要导入的多个项目，在 Location of NetBeans Projects（NetBeans 项目的位置）文本框中输入 NetBeans 项目位置，最后单击“完成”按钮，把选中的 Eclipse 项目导入到指定的 NetBeans 项目目录中。

在图 2-14 中，如果 Eclipse 项目不是在工作区中创建的，那么需要选中 Import Project ignoring Project Dependencies（导入项目时忽略项目依赖关系）单选按钮，然后在 Project to Import（要导入的项目）文本框中输入 Eclipse 项目的根目录，在 Destionation Folder（目标文件夹）中输入要创建的 NetBeans 项目文件夹（在该文件夹中将创建 NetBeans 必需的项目文件，例如 project.xml），如图 2-17 所示。

这里建议使用 Eclipse 项目文件夹作为目标文件夹，否则源代码和项目文件将会分别保存在两个不同的文件夹中。设置完成后，单击“完成”按钮即可导入 Eclipse 项目。

Download at http://www.pin5i.com/

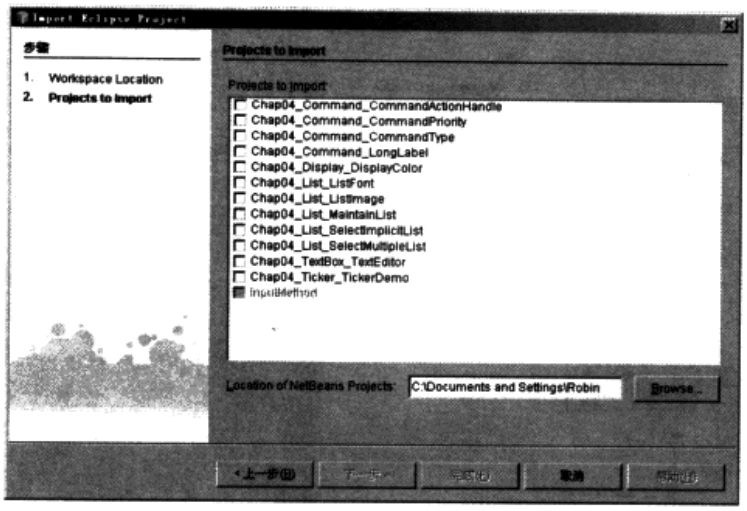


图 2-16 “Projects to Import”对话框

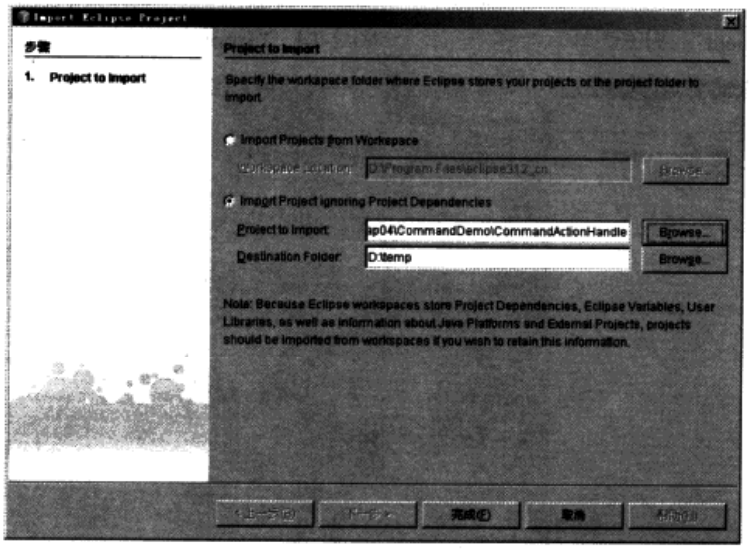


图 2-17 导入单独的项目文件

2.1.4 重命名项目

在项目创建后如果需要修改项目的名称，可以在“项目”窗口中右键单击项目的根节点，然后选择“重命名项目”命令，打开“重命名项目”对话框，如图 2-18 所示。

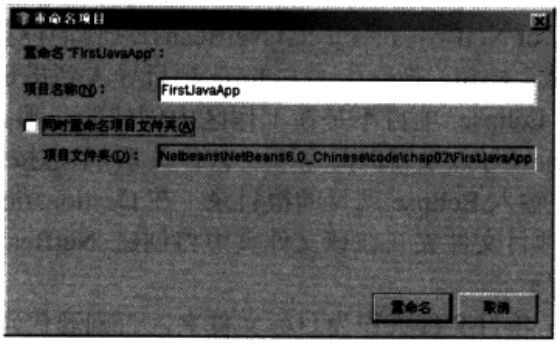


图 2-18 “重命令项目”对话框

Download at http://www.pin5i.com/

在“项目名称”文本框中输入项目的新名字。如果希望在重新命名项目的时候也同时修改项目所在的文件夹的名字，则选中“同时重命名项目文件夹”复选框。单击“重命名”按钮即可完成项目的重命名。

2.1.5 项目的复制、移动和删除

在项目创建后如果要复制项目，可以在“项目”窗口中右键单击项目的根节点，然后选择“复制项目”命令，打开“复制项目”对话框，如图 2-19 所示。

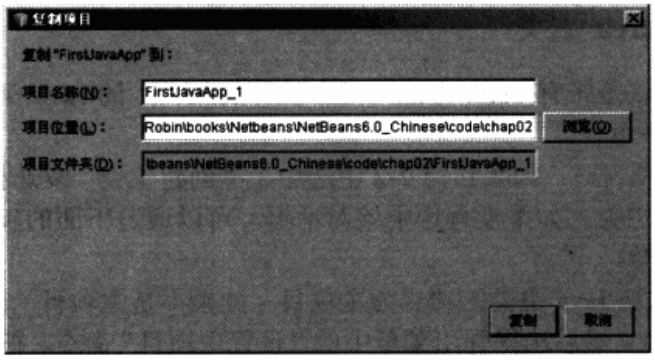


图 2-19 “复制项目”对话框

在“项目名称”文本框中输入新项目的名称；在“项目位置”文本框中输入项目的新位置或者单击“浏览”按钮查找项目的新路径，然后单击“复制”按钮，把当前选中的项目以指定的新名字复制到指定的目录。

在项目创建后如果要移动项目，可以在“项目”窗口中右键单击项目的根节点，然后选择“移动项目”命令，打开“移动项目”对话框，如图 2-20 所示。

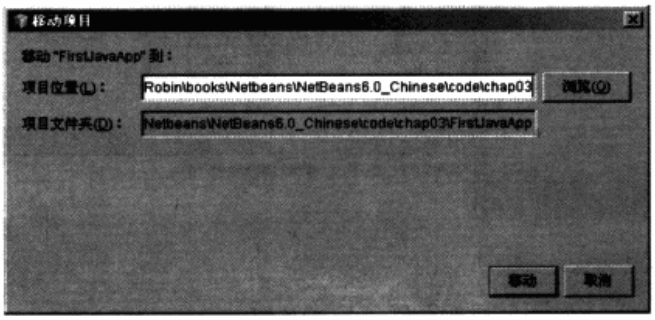


图 2-20 “移动项目”对话框

在“项目位置”文本框中输入项目的新位置或者单击“浏览”按钮查找项目的新路径，然后单击“移动”按钮完成项目的移动。

注意：*在将项目移动或复制到新位置时，新文件夹必须已存在于计算机上。如果不存在，可以在图 2-19 或者图 2-20 中使用“浏览”按钮来创建文件夹。*

在项目创建后如果要删除项目，可以在“项目”窗口中右键单击项目的根节点，然后选择“删除项目”命令，打开“删除项目”对话框，如图 2-21 所示。

如果希望删除项目的同时也删除源代码，那么选中图 2-21 中的复选框。确认无误后，单击“是”按钮就把该项目删除了。

Download at http://www.pin5i.com/

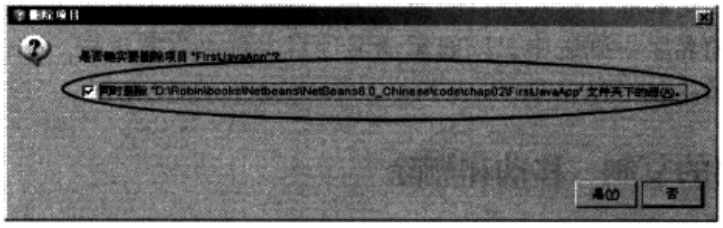


图 2-21 “删除项目”对话框

2.1.6 创建 Java 主类

所谓 Java 主类，就是带有 main 方法（程序执行的入口）的 Java 类，main 方法允许 Java 类作为控制台应用程序运行。

在创建 Java 标准项目时，如果在图 2-2 中选中了“创建主类”复选框，那么会在创建项目的同时创建一个 Java 主类。如果没有选中该对话框，可以通过下面的步骤在创建 Java 标准项目以后再创建 Java 主类。

❶ 首先确认要添加 Java 主类的项目为主项目。如果不是主项目，则在“项目”窗口中右键单击该项目的根节点，然后选择弹出菜单中的“设置主项目”命令，把该项目设置为主项目。

❷ 选择“文件”→“新建文件”命令，打开“新建文件”对话框，如图 2-22 所示。该对话框中提供了很多的模板，可以帮助开发人员快速地创建原型文件。

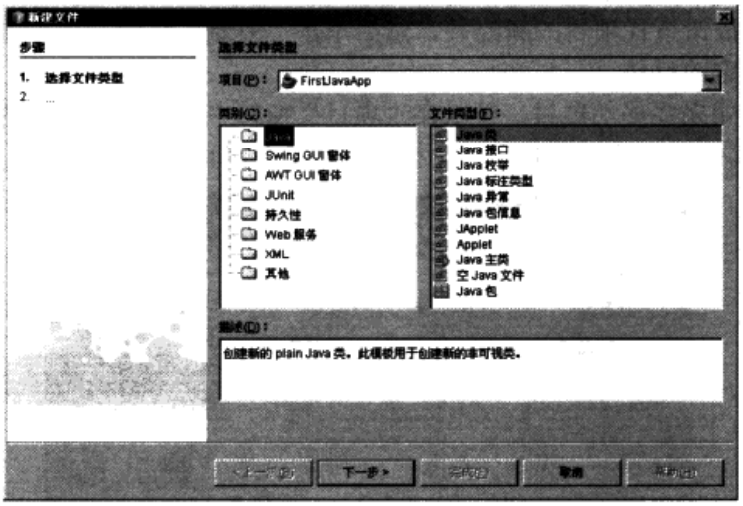


图 2-22 “新建文件”对话框

名词解释：模板

模板可以确定要创建对象的初始外观和行为。使用模板可以帮助开发人员节省用于创建应用程序的时间和精力。在“新建文件”对话框中，Java 组件作为标准模板提供，如 Swing 和“抽象窗口工具包（AWT）”容器。此外，还提供了用于 Applet、类、对话框、HTML 文件、文本文件和书签的模板。

❸ 在“类别”列表中，选择“Java”节点。在右边的“文件类型”列表中，会显示如下的模板供用户选择使用。

- Java 类：创建新的 plain Java 类。此模板用于创建新的非可视类。
- Java 接口：创建新的 Java 接口。

Download at http://www.pin5i.com/

- Java 枚举：创建新的 Java 枚举文件。要能够使用枚举文件，项目必须使用 JDK 5.0 或兼容版本。
- Java 标注类型：创建新的 Java 标注类型。要能够使用标注类型文件，项目必须使用 JDK 5.0 或兼容版本。
- Java 异常：创建带有可选详细消息的新 Exception 子类。
- Java 包信息：创建新的 Java 包信息。要使用包标注，项目必须使用 JDK 5.0 或兼容版本。
- JApplet：创建新的 JFC（Swing）Applet。Applet 是可以在任一启用了 Java 的浏览器上运行的 Java 类。注意：此模板不包含允许在窗体编辑器中对 Applet 进行可视化设计的窗体代码。要进行可视化设计，请选择“Swing GUI 窗体”类别下的 JApplet 模板。关于 JApplet 小程序的开发请参考第 6 章。
- Applet：创建新的 AWT（抽象窗口工具包）Applet。Applet 是可在任一启用了 Java 的浏览器上运行的 Java 类。注意：此模板不包含允许在窗体编辑器中对 Applet 进行可视化设计的窗体代码。要进行可视化设计，请从“Java GUI 窗体| AWT 窗体”类别下的 Applet 模板入手。关于 Applet 小程序的开发请参考第 6 章。
- 空 Java 文件：创建空的 Java 源文件。除了生成必需的包语句外，不生成任何代码。使用此模板可以从头开始创建类。
- Java 包：创建用于保存 Java 源文件的包。此包的实际形式为磁盘上的空文件夹。

❹ 这里，选择“文件类型”列表中的“Java 主类”节点，单击“下一步”按钮，打开“名称和位置”对话框，如图 2-23 所示。在“类名”文本框中输入 Java 主类的类名。在“包”下拉列表框中输入包的名字。

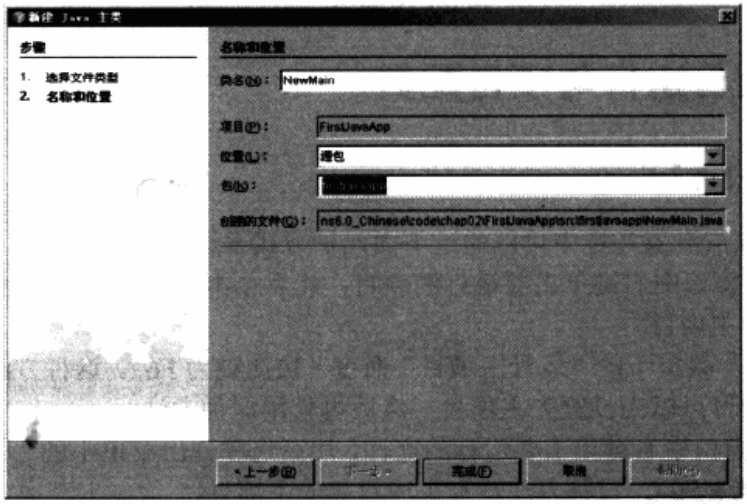


图 2-23 “名称和位置”对话框

❺ 单击“完成”按钮，即可创建一个 Java 主类。



2.1.7 创建 Java 包

NetBeans 不仅提供了创建 Java 类的功能，还提供了创建 Java 包的功能。可以通过下面的两种方式创建 Java 包。

- 在 NetBeans 的“项目”窗口中，右键单击要添加 Java 包的项目的“源代码包”节点或者“测试包”节点，然后选择弹出菜单的“新建”→“Java 包”命令，如图 2-24 所示。打开的“新建 Java 包”对话框如图 2-25 所示。

Download at http://www.pin5i.com/

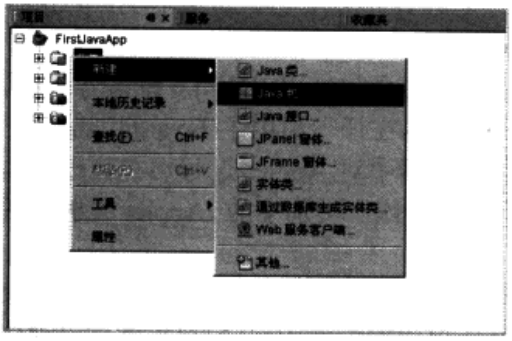


图 2-24 第 1 种添加 Java 包的方法

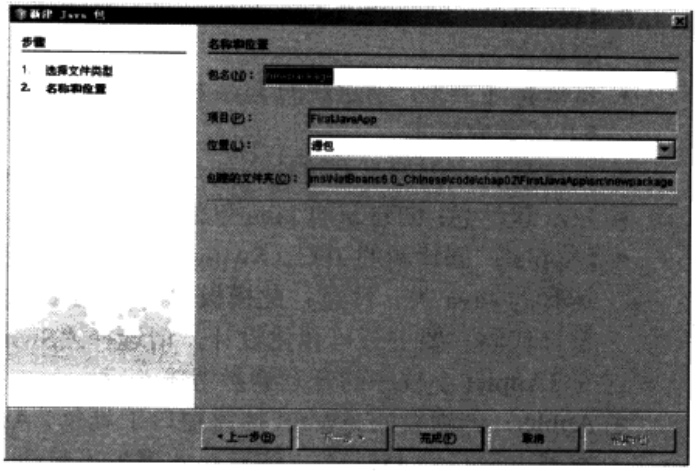


图 2-25 “新建 Java 包”对话框

- 把要添加 Java 包的项目设置为主项目，然后选择“文件”→“新建文件”命令，打开如图 2-22 所示的“新建文件”对话框。在“新建文件”对话框中选择“类别”列表中的“Java”节点，然后再选择“文件类型”列表中的“Java 包”节点。单击“下一步”按钮，打开如图 2-25 所示的“新建 Java 包”对话框。

在“新建 Java 包”对话框中可以输入 Java 包的名字和位置。在“包名”文本框中输入任意有效的名字，然后在“位置”下拉列表框中选择 Java 包的创建位置。单击“完成”按钮即可在指定的位置创建一个 Java 包。

2.2 项目运行与项目配置

2.2.1 项目运行

在 NetBeans 集成开发环境中可以通过下面几种方式来运行标准项目。

- 在“项目”窗口中右键单击要运行的项目，然后单击“运行项目”命令。这种方法适合主项目和非主项目。
- 选择“运行”菜单下的“运行主项目”命令（快捷键为 F6）。这种方法只适用于主项目，对于非主项目可以先设置为主项目，然后再使用这种方法。
- 在“项目”窗口中右键单击要运行的文件，然后选择弹出菜单中的“运行文件”命令（快捷键为 Shift + F6）。这种方法适合主项目和非主项目。
- 在“项目”窗口中选中要运行的文件，或者在“编辑器”窗口中激活要运行的文件，然后选择“运行”菜单下的“运行文件”→“运行 xxxx.java”命令（快捷键为 Shift + F6）。这种方法适合主项目和非主项目。
- 单击工具栏上的“运行主项目”图标 。这种方法只适用于主项目，对于非主项目可以先设置为主项目，然后再使用这种方法。

说明：如果只想编译某个文件，可以在“项目”窗口中右键单击要编译的文件，然后选择弹出菜单中的“编译文件”命令（快捷键为 F9）。

Download at http://www.pin5i.com/

2.2.2 项目配置

在 NetBeans 集成开发环境中，可以修改项目的配置参数，也可为项目指定另外的源代码目录、测试目录、指定编译选项、设置编译运行库、设置运行参数等。

1. 设置源代码选项

在“项目”窗口中右键单击项目，选择弹出菜单中的“属性”命令。或者在“项目”窗口中选中项目，然后选择“文件”菜单的“选中的项目名字属性”命令，打开“项目属性”对话框，在左边的“类别”列表中选中“源”节点，如图 2-26 所示。

NetBeans 的标准项目允许有多个源代码文件夹，在“源包文件夹”列表中列出了这些源代码文件夹（默认只有一个 src 文件夹）。可以单击旁边的“添加文件夹”按钮，添加其他的源代码文件夹。NetBeans 的标准项目也允许有多个测试代码文件夹，在“测试包文件夹”列表中列出了这些测试代码文件夹（默认只有一个 test 文件夹）。可以单击旁边的“添加文件夹”按钮，添加其他的测试代码文件夹。“源代码/二进制格式”下拉列表框用于选择 JDK 的版本。“编码”下拉列表框用于选择代码的编码格式。

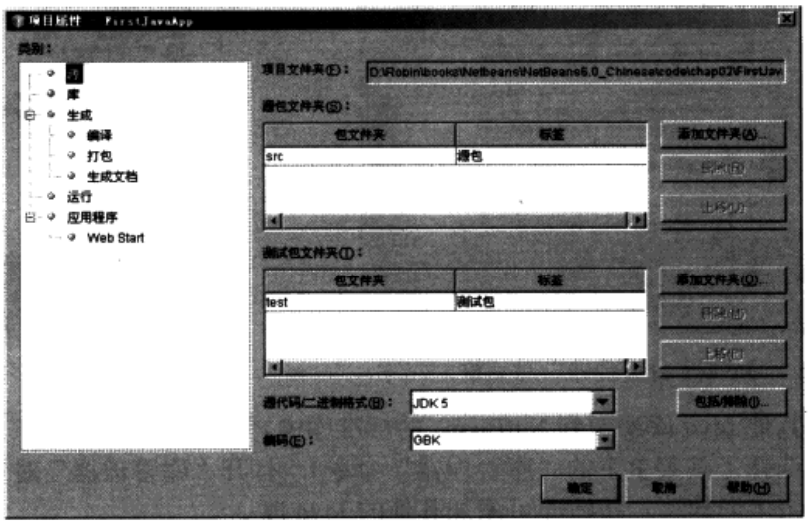


图 2-26 在“项目属性”对话框中修改源代码选项

2. 设置库选项

在“项目属性”对话框中，选中“类别”列表中的“库”节点，可以设置应用程序编译和运行时的库，如图 2-27 所示。如果在自己的项目中使用到了其他的 Java 库，或者引用了其他的项目，那么需要在这里添加相关的库文件。

“Java 平台”下拉列表框可以用来指定编译使用的 JDK 版本，如果要添加其他的 Java 平台，可以单击“管理平台”按钮（或者在 NetBeans 集成开发环境中选择工具菜单下的“Java 平台管理器”命令），打开“Java 平台管理器”对话框，如图 2-28 所示。在这个对话框中可以添加/删除 Java 平台。

如果项目中引用了其他项目，那么可以在如图 2-27 所示的对话框中单击“添加项目”按钮，引入其他的项目；如果需要添加一个单独的 Jar、Zip 或者一个包含.class 文件的文件夹，可以单击“添加 JAR/文件夹”按钮；如果要添加一个 NetBeans 库，可以单击“添加库”按钮，打开“添加库”对话框，如图 2-29 所示。

Download at http://www.pin5i.com/

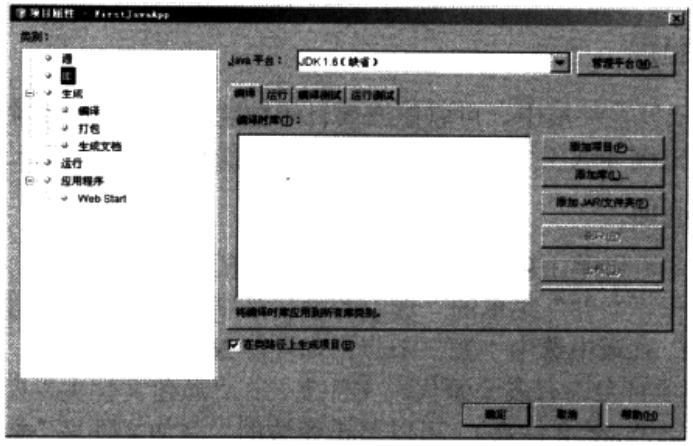


图 2-27 设置编译库

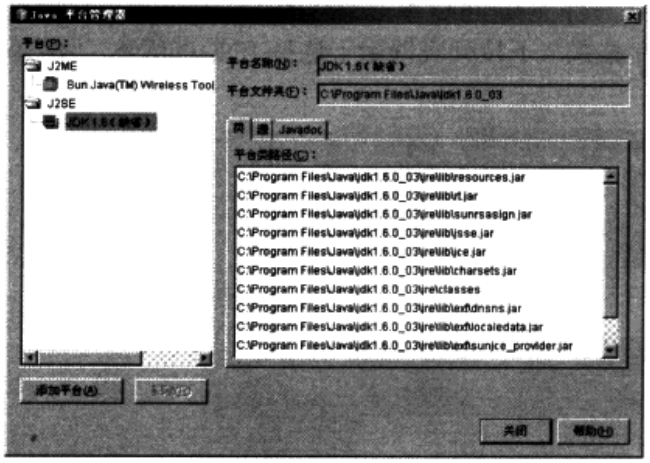


图 2-28 “Java 平台管理器”对话框

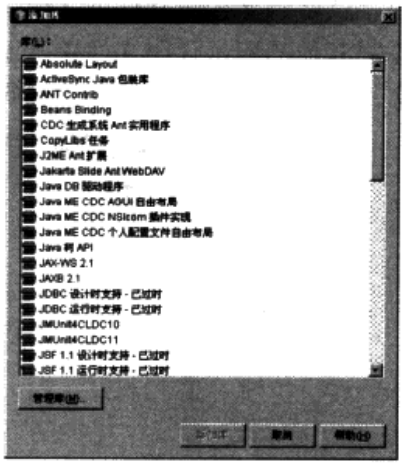


图 2-29 “添加库”对话框

用户可以把其他 Java 库添加到 NetBeans 的标准库中，单击“管理库”按钮（或者在 NetBeans 集成开发环境中选择工具菜单下的“库管理器”命令），打开“库管理器”对话框，如图 2-30 所示。这里，单击“新建库”按钮即可添加其他的 Java 库。

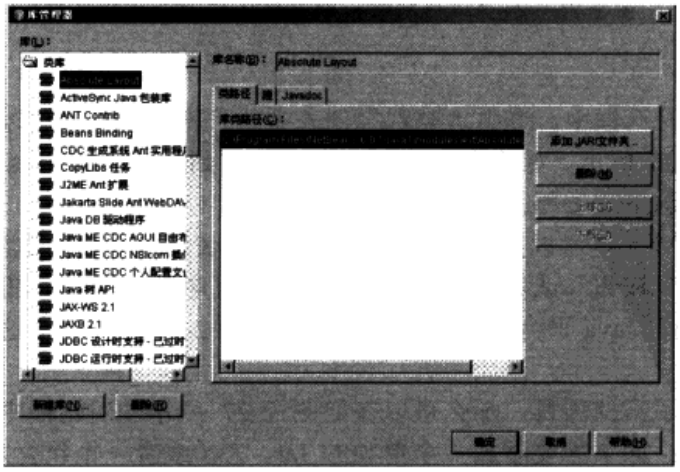


图 2-30 “库管理器”对话框

添加完毕后，单击“确定”按钮，返回“添加库”对话框，此时就可以选择新添加的 NetBeans 库了。在“添加库”对话框中选中要添加的库，然后单击“添加库”按钮，选中的库将被添加

Download at http://www.pin5i.com/

到“编译时库”列表中，如图 3-31 所示。

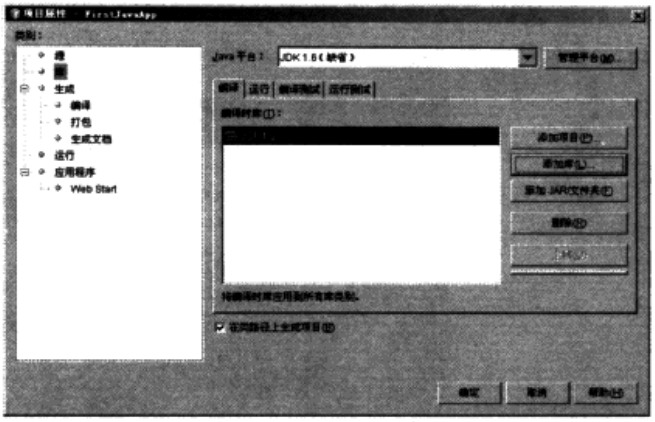


图 3-31 添加了一个编译时库

使用同样的方法可以在“运行”标签中添加运行时必需的库；在“编译测试”标签中添加编译测试程序时必需的库；在“运行测试”标签中添加运行测试程序时必需的库。

3. 设置编译选项

在“项目属性”对话框中，选中“类别”列表中的“生成”→“编译”节点，可以设置应用程序编译选项，如图 2-32 所示。

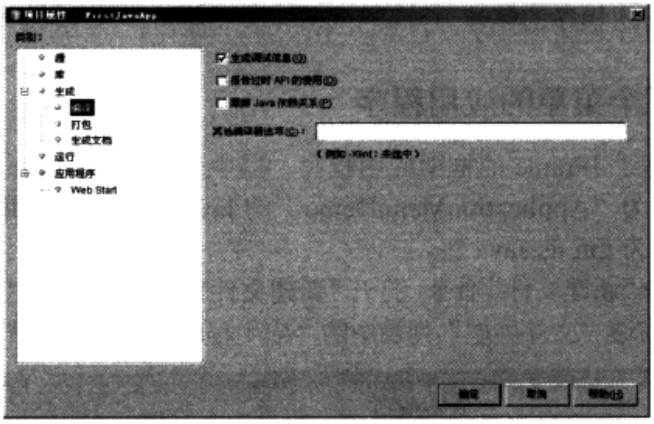


图 2-32 设置应用程序的编译选项

选中“生成调试信息”复选框，在编译的文件中会包含调试信息，可以进行跟踪调试。在开发和调试阶段请选中该复选框。全部测试通过后，发布正式的应用程序时，可以去掉这个选项。

选中“报告过时 API 的使用”复选框，当程序中使用了当前 JDK 已经不再建议使用的 API（即过时的 API）时，那么就会提示给开发人员。

如果要在编译时制定其他编译选项，那么可以在“其他编译器选项”文本框中输入。关于编译器的选项，因为超出了本书的介绍范围，这里不再介绍，请读者自行参考相关文档。

4. 设置运行选项

在“项目属性”对话框中，选中“类别”列表中的“运行”节点，可以设置应用程序的运行选项，如图 2-33 所示。

在“主类”文本框中可以输入 Java 程序运行的主类，如果项目中有多个主类，这里可以单击“浏览”按钮，打开“浏览主类”对话框，选择哪个类要被应用程序执行。

在“参数”文本框中可以输入应用程序运行时需要的一些参数。“工作目录”文本框用于

Download at http://www.pin5i.com/

制定应用程序的工作目录。“VM 选项”文本框用于指定虚拟机的一些选项。关于虚拟机的具体选项，请读者自行查阅相关文档，这里不再赘述。

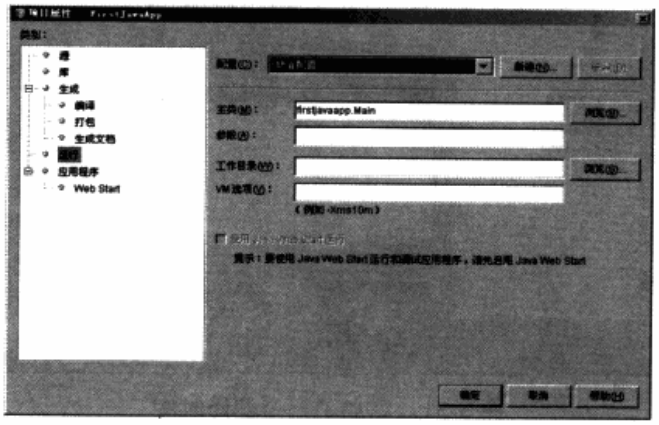


图 2-33 设置应用程序的运行选项

2.3 创建标准 GUI 应用程序

通过 NetBeans 集成开发环境提供的模板可以快速地创建带有菜单的应用程序、多文档界面（Multi-Document Interface，简称 MDI）应用程序和包含“确定/取消”对话框的应用程序。下面分别介绍这几种应用程序的创建方法。

2.3.1 包含三个菜单的应用程序

下面将创建一个基于 JFrame 的框架应用程序，其中包含 File、Edit 和 Help 三个菜单。

❶ 新建一个名字为“ApplicationMenuDemo”的 Java 标准项目，并设置为主项目。

❷ 新建一个名字为 gui 的 Java 包。

❸ 选择“文件”→“新建文件”命令，打开“新建文件”对话框。选择“类别”列表中的“Swing GUI 窗体”，然后选择右边“文件类型”列表中的“应用程序样例窗体”节点，如图 2-34 所示。

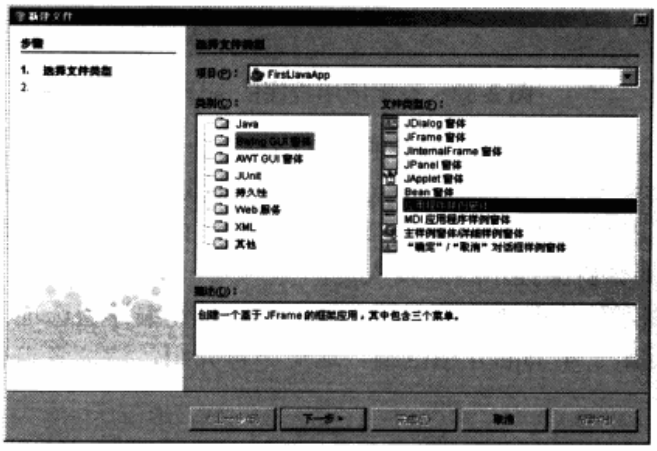


图 2-34 新建样例应用程序

❹ 单击“下一步”按钮，打开“新建应用程序样例窗体”对话框。在对话框中输入类名，并选择创建到 gui 包中，如图 2-35 所示。

Download at http://www.pin5i.com/

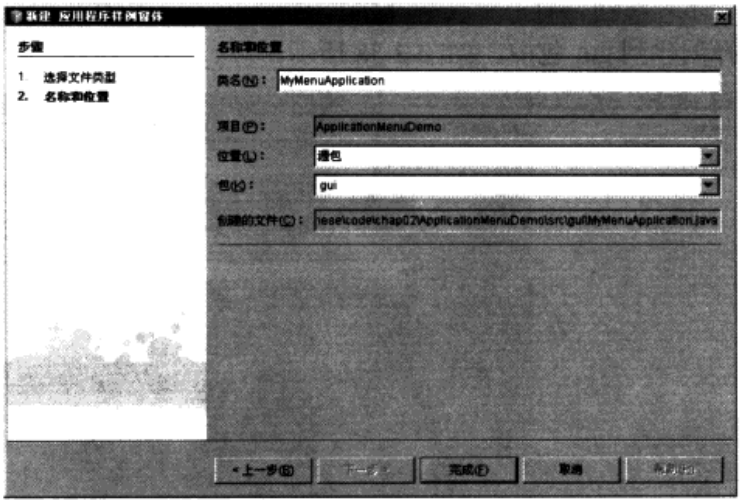


图 2-35 “新建应用程序样例窗体”对话框

❺ 单击“完成”按钮，创建带菜单的应用程序。创建完成后，可以在“设计”视图中看到应用程序上面有三个菜单，如图 2-36 所示。

❻ 运行程序会显示如图 2-37 所示的应用程序。当选择“File”→“Exit”命令时会退出应用程序，其他菜单项因为没有添加代码，所以单击时没有反应。

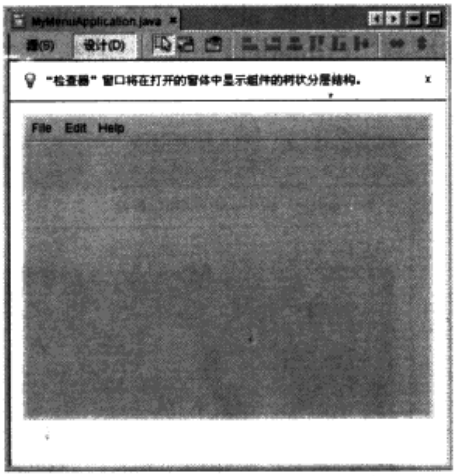


图 2-36 “设计”视图中的应用程序菜单

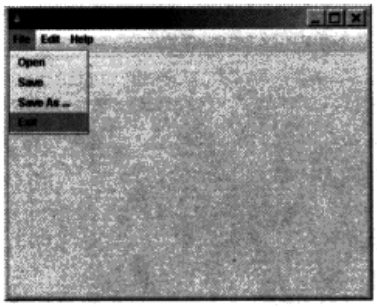


图 2-37 创建的应用程序运行界面

2.3.2 多文档界面应用程序

多文档界面（Multiple Document Interface, MDI）应用程序是一种比较常见的应用程序，利用 NetBeans 提供的模块可以快速地创建一个多文档界面应用程序。

多文档界面模型类似于传统的计算机窗口系统，它也有一个包含其他窗口的桌面。在基于 JFC/Swing 的多文档界面应用程序中，各个内部窗口都包含在一个封闭窗口（即桌面）中，用户可以对该窗口定位、调整大小、最小化和关闭。

NetBeans 提供了创建多文档界面应用程序的模板，可以像创建带菜单的应用程序一样，快速地生成一个带三个菜单的多文档界面应用程序。下面介绍具体的开发步骤。

❶ 新建一个名字为“MDIAppDemo”的 Java 标准项目，并设置为主项目。

❷ 新建一个名字为 gui 的 Java 包。

❸ 在如图 2-34 所示的对话框中选择“文件类型”列表中的“MDI 应用程序样例窗体”节

Download at http://www.pin5i.com/

点，然后单击“下一步”按钮，打开“新建 MDI 应用程序样例窗体”对话框。在对话框中输入类的名字，并选择创建到 gui 包中，如图 2-38 所示。

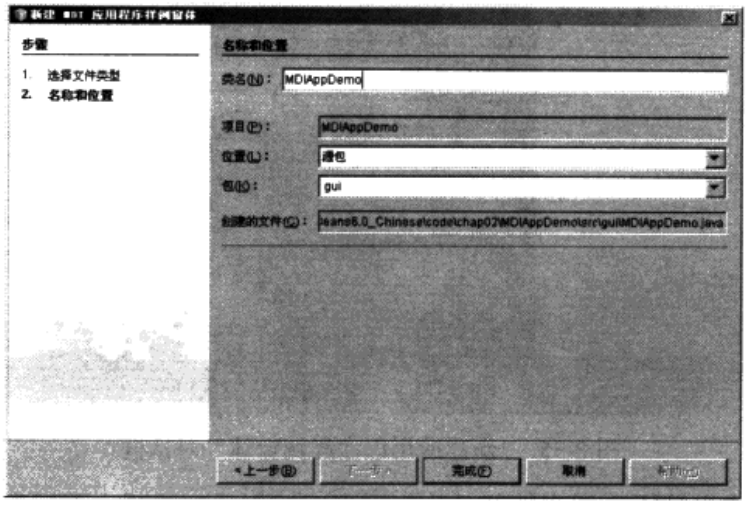


图 2-38 “新建 MDI 应用程序样例窗体”对话框

❹ 单击“完成”按钮，创建带菜单的多文档界面应用程序。创建完成后，可以在“设计”视图中看到应用程序上面有三个菜单，如图 2-41 所示。

❺ 此时，只是创建了一个多文档界面程序的框架，需要再添加内部窗体（JInternalFrame）。从“组件面板”窗口的“Swing”类别中选择“内部框架”（JInternalFrame）组件，将其拖曳到图 2-39 的蓝色区域中，如图 2-40 所示。

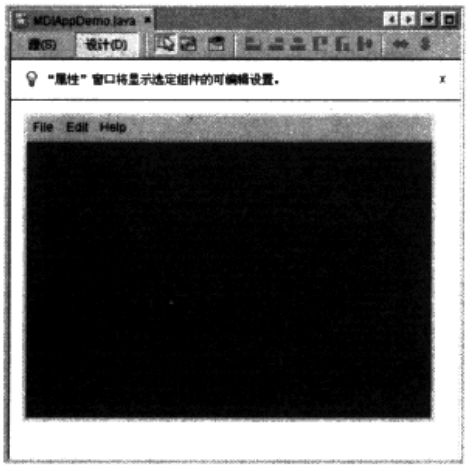


图 2-39 空白的多文档界面应用程序

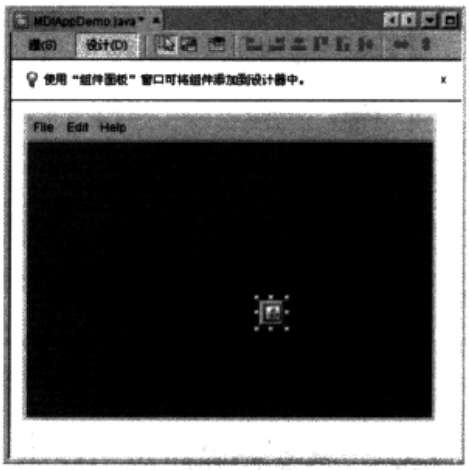


图 2-40 添加内部窗体

❻ 通过鼠标调整内部窗体的大小和位置，如图 2-41 所示。

❼ 运行程序会显示如图 2-42 所示的应用程序运行效果。当选择“File”→“Exit”命令时会退出应用程序，其他菜单项因为没有添加代码，所以单击时没有反应。至此，完成了一个简单的多文档界面应用程序。



2.3.3 “确定”/“取消”对话框

MetBeans 提供了快速创建对话框的模板，下面的操作将创建一个带有“确定”和“取消”按钮的 Swing 对话框。

Download at http://www.pin5i.com/

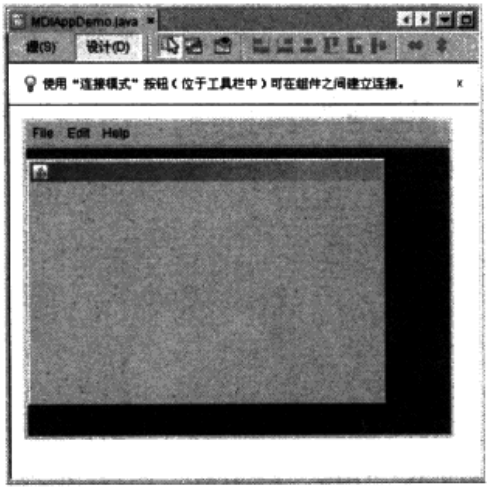


图 2-41 调整内部窗体的大小和位置

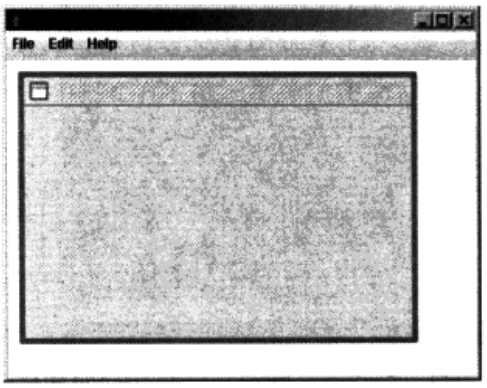


图 2-42 多文档界面应用程序运行效果

❶ 新建一个名字为“OkCancelDialogDemo”的 Java 标准项目，并设置为主项目。

❷ 新建一个名字为 gui 的 Java 包。

❸ 在如图 2-34 所示的对话框中选择“文件类型”列表中的“‘确定’/‘取消’对话框样例窗体”，然后单击“下一步”按钮，打开“新建‘确定’/‘取消’对话框样例窗体”对话框。在对话框中输入类的名字，并选择创建到 gui 包中，如图 2-43 所示。

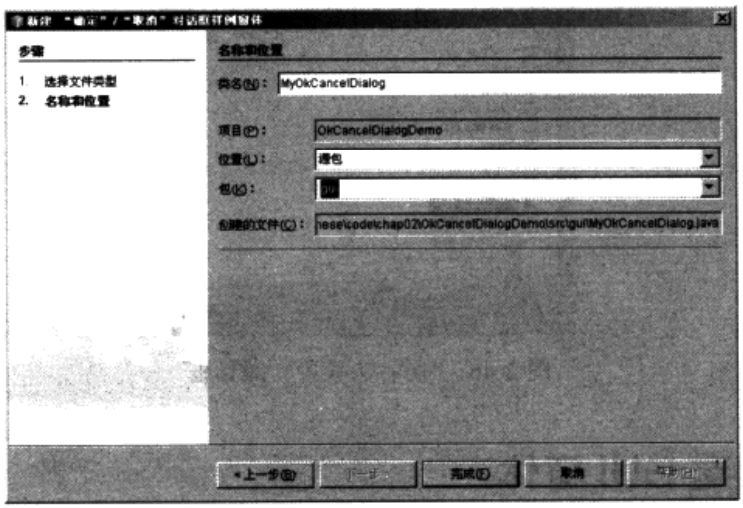


图 2-43 “新建‘确定’/‘取消’对话框样例窗体”对话框

❹ 单击“完成”按钮，创建一个对话框，在对话框上有一个 OK 按钮和一个 Cancel 按钮，如图 2-44 所示。程序的运行界面如图 2-45 所示，单击 OK 按钮或者 Cancel 按钮时，都会自动关闭对话框。

提示：可以使用上面的模板创建消息框和任何其他对话框模板（例如，创建一个带有“是”、“否”和“取消”按钮的模板）。首先创建对话框，然后在“项目”窗口中右键单击源文件节点，从上下文菜单中选择“另存为模板”，打开“另存为模板”对话框，如图 2-46 所示。在该对话框中选择模板的类别，然后单击“确认”按钮即可。下次再打开如图 2-34 所示“新建文件”对话框时，就可以使用自己创建的模板了。

Download at http://www.pin5i.com/



图 2-44 创建的“确定”/“取消”对话框

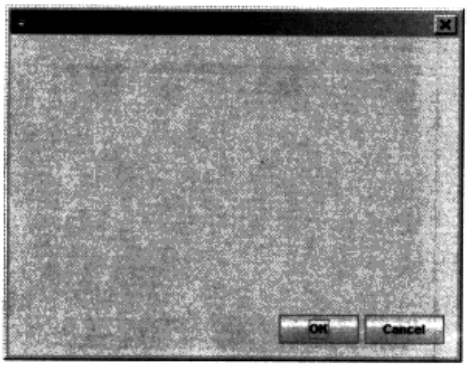


图 2-45 “确定”/“取消”对话框的运行效果

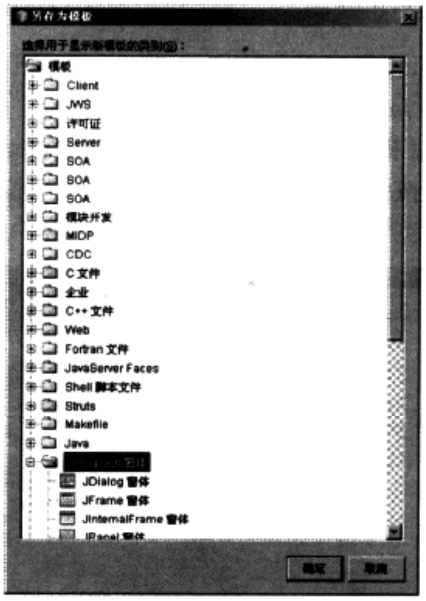


图 2-46 “另存为模板”对话框

2.4 创建 GUI 窗体

利用 NetBeans 提供的模板，可以快速地开发基于图形用户界面的窗体。NetBeans 中支持创建如下几个类型的 GUI 窗体：

- Java 小程序窗体
- 对话框窗体
- 框架窗体
- 内部框架窗体
- 面板窗体
- Bean 窗体

其中，Java 小程序窗体将在第 6 章进行介绍，Bean 窗体将在第 7 章进行介绍，内部框架窗体已在第 2.3 节进行了介绍。其余的窗体创建方法将在下面进行详细的讨论。

Download at http://www.pin5i.com/

2.4.1 对话框窗体

对话框是模型化或非模型化的窗口，通常用于提示用户输入信息。NetBeans 提供了创建基于 AWT（抽象窗口工具包）的对话框模板（Dialog）和基于 JFC（Swing）的对话框（JDialog）模板。下面以创建 JDialog 为例，介绍对话框窗体的创建方法。

❶ 新建一个 Java 项目 JDialogDemo，并设置为主项目。

❷ 新建一个名字为 gui 的 Java 包。

❸ 打开如图 2-34 所示的“新建文件”对话框。在“类别”列表中选择“Swing GUI 窗体”节点，然后在“文件类型”列表中选择“JDialog 窗体”节点。

说明：如果要创建基于 AWT 的窗体，这里应选择“类别”列表中的“AWT GUI 窗体”节点，然后再选择“文件类型”列表中的“对话框窗体”节点。

❹ 单击“下一步”按钮，打开“新建 JDialog 窗体”对话框，如图 2-47 所示。在“类名”文本框中输入窗体的类名；在“包”下拉列表框中选择新建的对话框窗体要被包含在哪个包中，这里选择 gui 包。单击“完成”按钮，即可创建 JDialog 窗体。

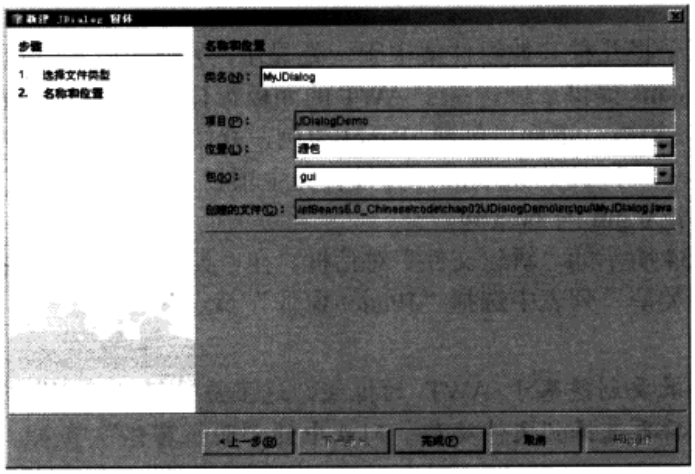


图 2-47 “新建 JDialog 窗体”对话框

2.4.2 框架窗体

框架通常用作独立式顶层窗口，承担应用的主用户界面。NetBeans 提供了创建基于 AWT 的框架模板（Frame）和基于 JFC（Swing）的框架（JFrame）模板。下面以创建 JFrame 为例，介绍框架窗体的创建方法。

❶ 新建一个 Java 项目 JFrameDemo，并设置为主项目。

❷ 新建一个名字为 gui 的 Java 包。

❸ 打开如图 2-34 所示的“新建文件”对话框。在“类别”列表中选择“Swing GUI 窗体”节点，然后在“文件类型”列表中选择“JFrame 窗体”节点。

说明：如果要创建基于 AWT 的框架，这里应选择“类别”列表中的“AWT GUI 窗体”节点，然后再选择“文件类型”列表中的“框架窗体”节点。

❹ 单击“下一步”按钮，打开“新建 JFrame 窗体”对话框，如图 2-48 所示。在“类名”

Download at http://www.pin5i.com/

文本框中输入框架窗体的类名；在“包”下拉列表框中选择新建的框架窗体要被包含在哪个包中，这里选择 gui 包。单击“完成”按钮，即可创建 JFrame 窗体。

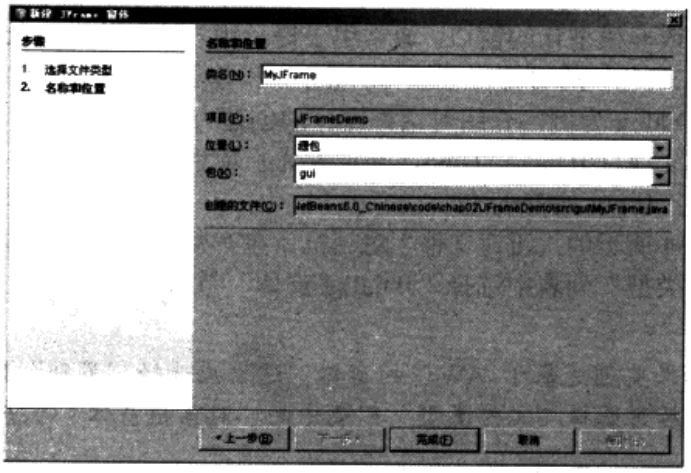


图 2-48 “新建 JFrame 窗体”对话框

2.4.3 面板窗体

面板窗体通常用于将几种其他组件集中于一处显示，另外面板本身也作为框架和对话框的内部组件使用。NetBeans 提供了创建基于 AWT 的面板窗体（Panel）模板和基于 JFC（Swing）的面板窗体（JPanel）模板。下面以创建 JPanel 为例，介绍面板窗体的创建方法。

❶ 新建一个 Java 项目 JPanelDemo，并设置为主项目。

❷ 新建一个名字为 gui 的 Java 包。

❸ 打开如图 2-34 所示的“新建文件”对话框。在“类别”列表中选择“Swing GUI 窗体”节点，然后在“文件类型”列表中选择“JPanel 窗体”节点。

说明：如果要创建基于 AWT 的框架，这里应选择“类别”列表中的“AWT GUI 窗体”节点，然后再选择“文件类型”列表中的“面板窗体”节点。

❹ 单击“下一步”按钮，打开“新建 JPanel 窗体”对话框，如图 2-49 所示。在“类名”文本框中输入框架窗体的类名；在“包”下拉列表框中选择新建的框架窗体要被包含在哪个包中，这里选择 gui 包。单击“完成”按钮，即可创建 JPanel 窗体。

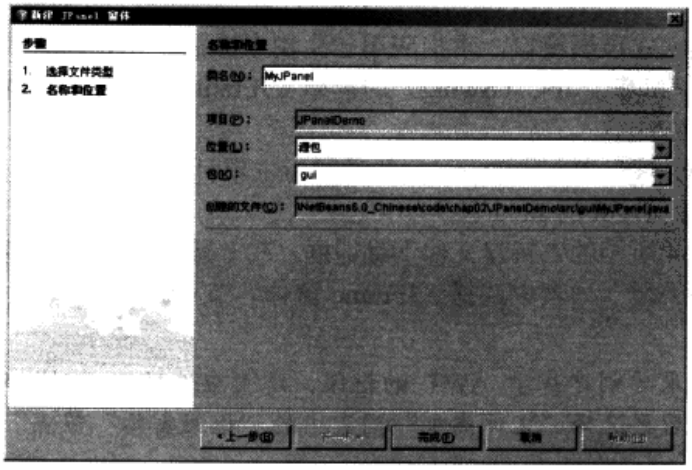


图 2-49 “新建 JPanel 窗体”对话框

Download at http://www.pin5i.com/

2.5 添加界面组件

NetBeans 集成开发环境的图形用户界面（Graphical User Interfaces，GUI）设计器，即 GUI Builder，可以可以通过可视方式来设计 GUI。在创建和修改 GUI 时，集成开发环境会自动生成用于实现界面的 Java 代码。从 NetBeans 5.0 开始，GUI Builder 已经进行了改进，变得更加直观，功能也更加强大。即使用户不熟悉布局管理器，也能轻松地构建出专业级的图形用户界面应用程序。NetBeans 的新 GUI Builder 允许简单地将组件置于需要它们的位置上来布置窗体。

2.5.1 GUI Builder

NetBeans 集成开发环境的 GUI Builder 通过简化图形界面解决了 Java GUI 创建的核心问题，从而把开发人员从 Swing 布局管理器的复杂性中解放出来。GUI 提供了简单的自由设计模式，使得界面组件的布局可以轻松完成。

布置窗体时，GUI Builder 将提供可视化的准线，给出最佳间距和组件对齐建议。在后台，GUI Builder 将设计决策转换为可用的 UI，这种 UI 是使用新 GroupLayout 布局管理器及其他 Swing 构造实现的。因为使用的是动态布局模型，所以使用 GUI Builder 构建的图形界面能通过调整以适应任何更改，而不改变已定义的组件之间的关系。只要调整窗体大小、切换地点或指定不同外观，GUI 就会按照目标外观的插入和偏移自动进行调整。

在“项目”窗口中单击包含 GUI 窗体的源代码，可以打开 GUI Builder 编辑窗体，如图 2-50 所示。

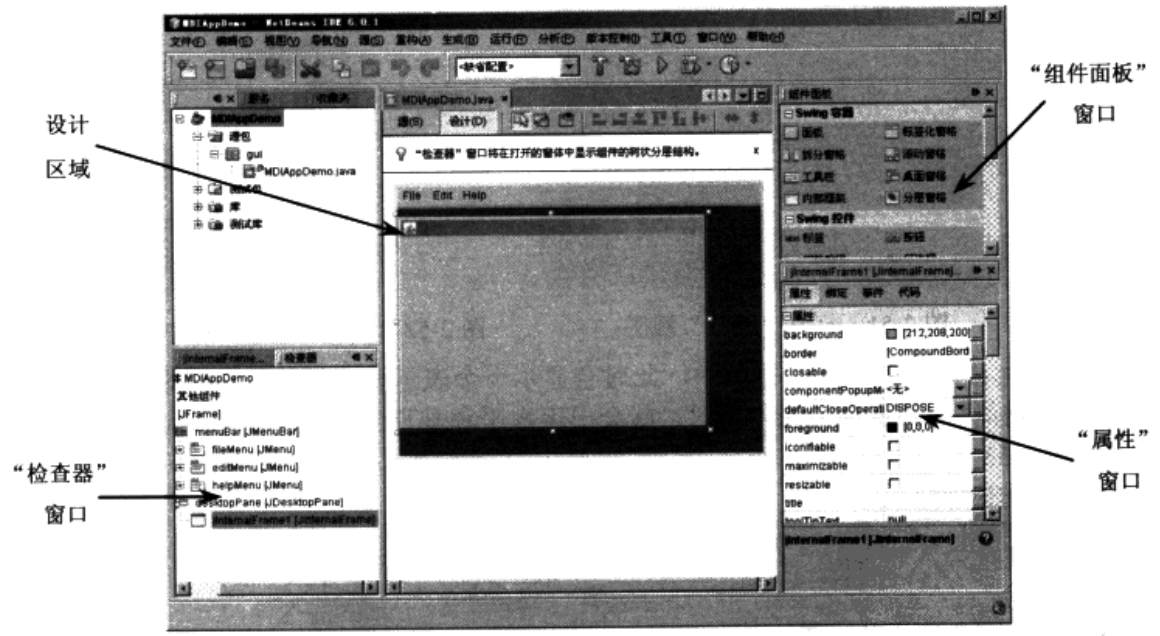


图 2-50 GUI Builder 编辑窗体

GUI Builder 不是一个单独的窗口，而是一组窗口的集合，通过这个集合可以快速地、可视化地设计窗体。GUI Builder 的各种窗口介绍如下。

- 设计区域：用于创建和编辑 Java GUI 窗体的 GUI Builder 的主窗口，由“设计”视图和“源代码”视图组成，分别用于查看类的源代码及其 GUI 组件的图形视图。工具栏的“源代码”和“设计”切换按钮，允许在这两个视图间进行切换。在设计区域中还包含其他工具栏按钮，提供常见命令的快捷访问，如在对齐组件、设置组件自动调整大小行为及预览窗体等。

Download at http://www.pin5i.com/

- “检查器”窗口（Inspector）：按照树层次结构提供应用程序中所有组件（可视的和非可视的）的表示形式。“检查器”窗口还提供关于在 GUI Builder 中当前正在编辑树中的什么组件的可视化反馈，同时还允许在可用面板中组织组件。
- “组件面板”窗口：组件面板中的组件可以被添加到 GUI 窗体中。
- “属性”窗口：显示并修改在“设计”视图、“检查器”窗口、“项目”窗口或“文件”窗口中当前选择组件的属性。

默认情况下，GUI 生成器使用 GroupLayout 布局管理器来管理新创建的所有窗体。GroupLayout 是 Netbeans 中新引入的一个布局管理器，它将成为 JDK 6 中 Swing 的核心部分。下面将主要介绍如何为 GroupLayout 添加组件，关于其他布局管理器将在第 2.5.5 节中进行介绍。



2.5.2 添加/编辑组件

NetBeans 的 GUI Builder 可以轻松地为窗体添加 Swing、AWT 或者 Java Bean 组件，并且设置组件之间的间距、大小和对齐方式。下面通过一个具体的例子演示如何为窗体添加和编辑组件。

❶ 按照第 2.1.5 节中的项目复制方法，把第 2.3.3 节中创建的对话框应用程序复制并保存为 ComponentsDemo，并把创建的 ComponentsDemo 项目设置为主项目。

❷ 在“项目”窗口中，双击 ComponentsDemo→“源代码包”→gui→MyOkCancelDialog.java 节点，打开“设计”视图，如图 2-51 所示。

❸ 选中“组件面板”窗口中的“面板”（JPanel）组件，如图 2-52 所示。

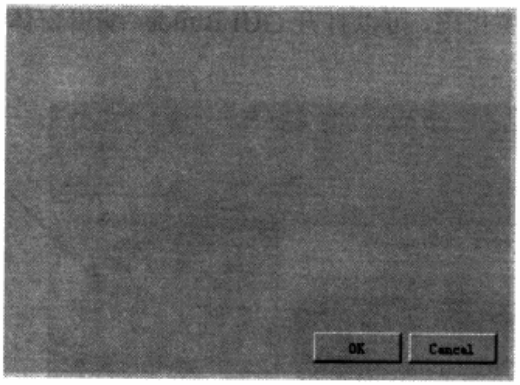


图 2-51 打开“设计”视图

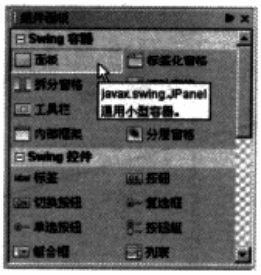


图 2-52 在“组件面板”中选中 JPanel 组件

❹ 把鼠标移动到“设计”视图中，此时会显示一个表示要添加的组件矩形框，并随着鼠标移动。当鼠标移动到窗体边缘附近时，就会出现水平和垂直方向的两个校准线，指示较佳的边距，如图 2-53 所示。

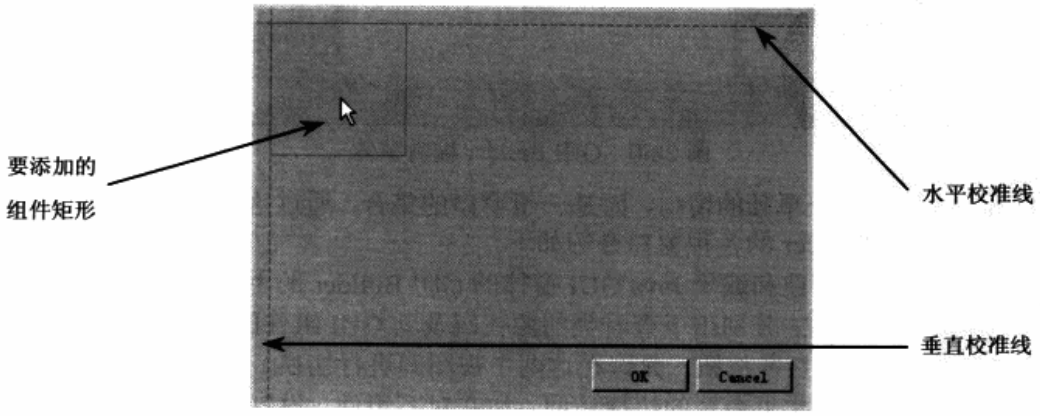


图 2-53 在窗体中添加 JPanel 组件

Download at http://www.pin5i.com/

❺ 把组件移动到目标位置后，单击鼠标放置组件，如图 2-54 所示。此时在“检查器”窗口中就会多了一个面板组件，表示组件添加成功，如图 2-55 所示，

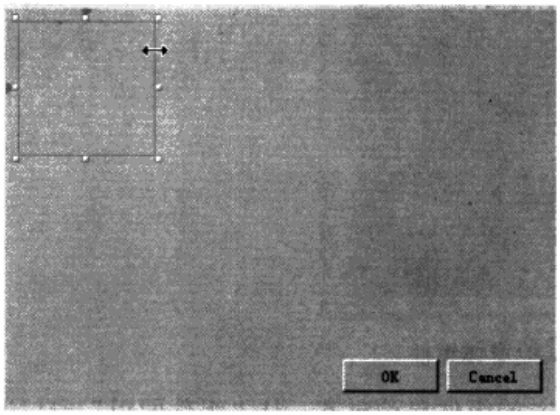


图 2-54 放置 JPanel 组件

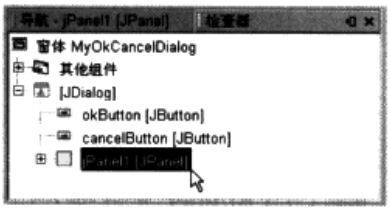


图 2-55 “检查器”窗口

❻ 单击 OK 按钮，此时面板会在窗体中消失，当鼠标移动到 JPanel 的位置时会显示一个灰色的矩形框，表示 JPanel 组件未被选中，如图 2-56 所示。单击面板中的任何一个位置，面板组件会被重新选中。在 GUI Builder 中，只有被选中的组件才能进行改变大小、对齐等操作。

❼ 下面调整组件的大小，因为要在上面添加其他的 GUI 组件。被选中的组件周围会显示带八个白色矩形的边框。当把鼠标移动到边框上或者白色矩形上时，鼠标的指针会改变。拖曳边框或者拖曳白色的矩形就会修改组件的大小，如图 2-57 所示。

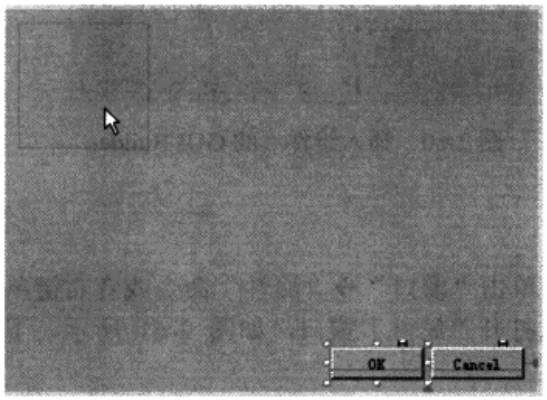


图 2-56 未被选中时的 JPanel 组件

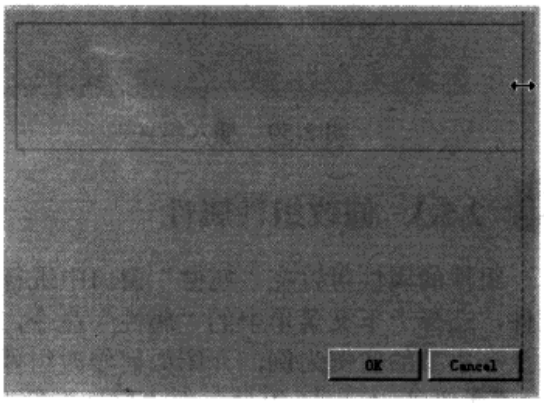


图 2-57 修改组件的大小



说明： NetBeans 的 GUI Builder 支持对一个容器单独进行设计（避免互相影响）。在要进行单独设计的组件上单击右键，选择弹出菜单中的“设计此容器”命令，就会在“设计”视图中只显示该容器。在设计完成后，再选择组件的上下文菜单的“设计顶层容器”命令，就可以返回上一级容器了。

❽ 重复上面的步骤，添加 4 个标签（JLabel）组件、两个文本字段（JTextField）组件和一个组合框（JComboBox）组件，如图 2-58 所示的组件。



提示： GUI Builder 也支持复制（Ctrl+C）、粘贴（Ctrl+V）和剪切（Ctrl+X）组件的操作，按下 Del 键以可以删除选中的组件。

Download at http://www.pin5i.com/

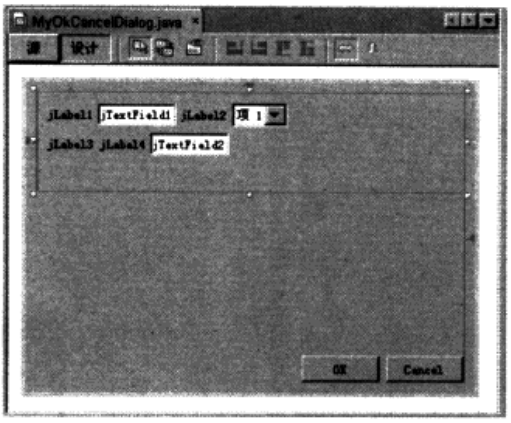


图 2-58 添加其他窗体组件

❾ GUI Builder 允许插入组件，并且会对插入的组件自动对齐。从“组件面板”窗口选中一个文本字段组件，把它拖曳到如图 2-59 所示的位置并松开鼠标，此时新添加的文本字段组件会插入在当前位置，如图 2-60 所示。

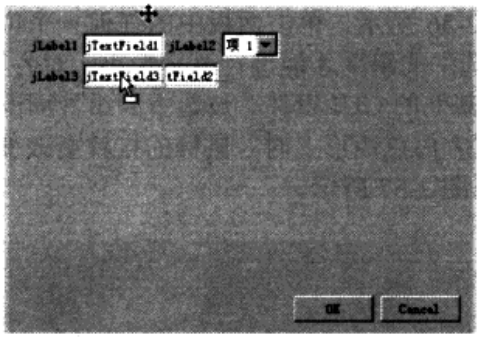


图 2-59 插入组件

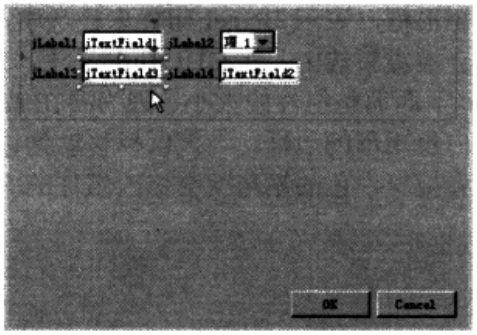


图 2-60 插入组件后的 GUI Builder



2.5.3 修改组件属性

组件的属性可以在“属性”窗口中进行编辑。单击“窗口”→“属性”命令或者右键单击组件，选择上下文菜单中的“属性”命令，都可以打开“属性”窗口，如图 2-61 所示。下面以上一小节的实例为例，介绍如何修改组件的属性。

❶ 选中 jLabel1 组件，在“属性”窗口中修改 text 属性为“姓名：”，或者双击 jLabel1 组件，直接输入“姓名：”，如图 2-62 所示。



图 2-61 “属性”窗口

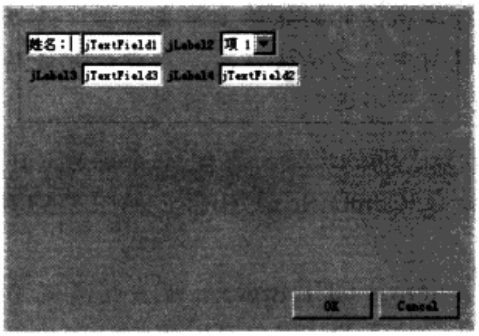


图 2-62 修改 text 属性

Download at http://www.pin5i.com/

❷ 单击“属性”窗口中 foreground 属性旁边的 ... 按钮，打开 ColorEditor 对话框，如图 2-63 所示。在 ColorEditor 对话框中选择前景颜色，显示效果可以在“预览”分组框中进行预览。ColorEditor 对话框中提供多种选择颜色的方法，也可以直接设置 RGB 颜色值、选择 AWT/Swing 组件面板中的颜色，或者选择组件面板中的颜色。

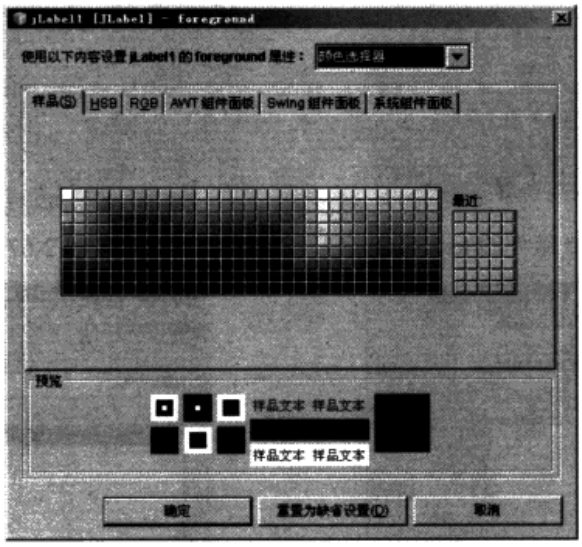


图 2-63 ColorEditor 对话框

❸ 这里选择红色，然后单击“确定”按钮。此时，第一个 jLabel 的颜色会变为如图 2-64 所示。

❹ 用同样的方法，把 jLabel2 的 text 属性设置为“性别：”，foreground 颜色也设置为红色。把 jLabel3 的 text 属性设置为“生日：”，把 jLabel4 的 text 属性设置为“工作单位：”，如图 2-65 所示。

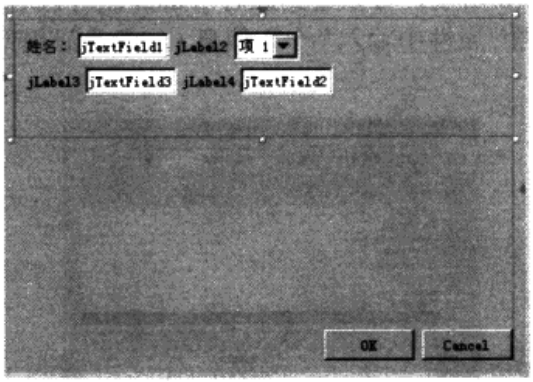


图 2-64 设置 jLabel 组件的前景颜色

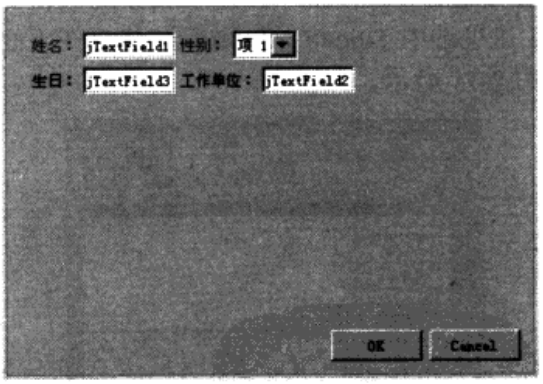


图 2-65 设置其他 jLabel 组件的属性

❺ 把 jTextField1、jTextField2 和 jTextField3 的 text 属性设置为空，此时这三个组件的大小会有所变化。把鼠标移动到组件的边框，并拖动鼠标，此时组件的大小会随着变化，如图 2-66 所示。用同样的方法，调整其他几个 jTextField 组件的大小，如图 2-67 所示。

❻ 选择下拉列表框 jComboBox 组件，在“属性”窗口中选择 model 属性旁边的 ... 按钮，打开 model 对话框，如图 2-68 所示。

model 对话框用于设置下拉列表框中显示的项目，这里删除两个项目，并修改剩下的两个项目，如图 2-69 所示。

Download at http://www.pin5i.com/

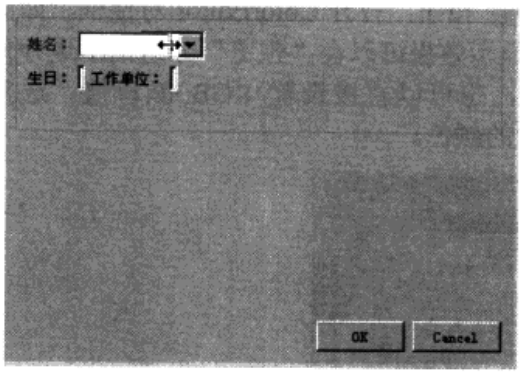


图 2-66 设置 jTextField 组件的 text 属性

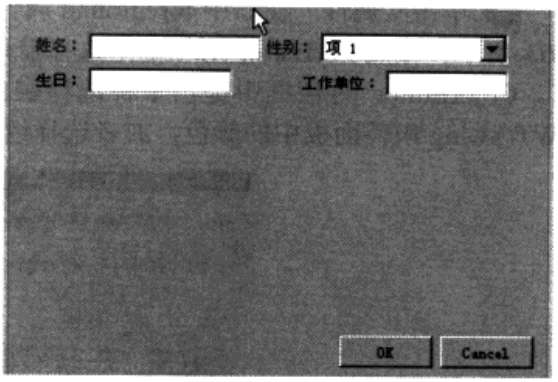


图 2-67 调整组件的大小

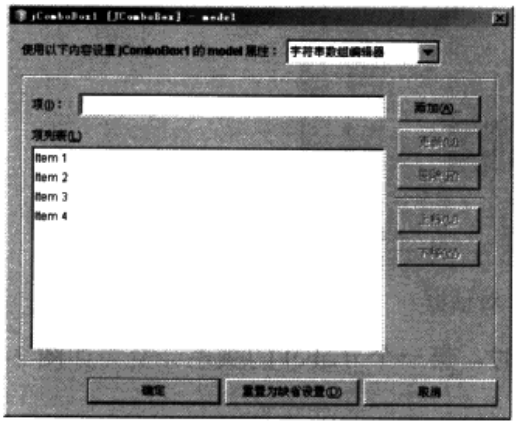


图 2-68 model 对话框

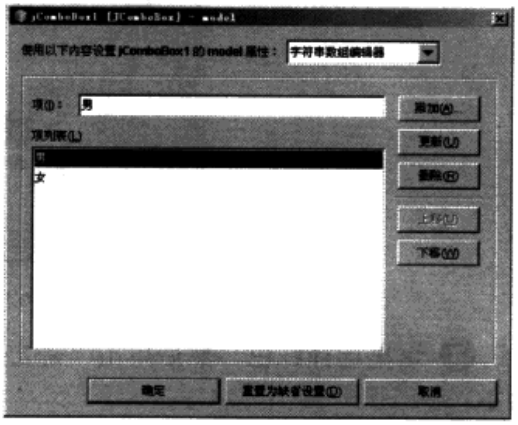


图 2-69 修改项目

❼ 单击“确定”按钮，返回 GUI Builder。选中 JPanel 组件，选择“属性”窗口中 border 属性旁边的 ··· 按钮，打开 border 对话框，如图 2-70 所示，可以看到此时 jPanel 组件没有边框。这里选中 TitledBorder 选项，然后在下面的“标题”属性中输入“个人信息（红色为必填）”，如图 2-71 所示。

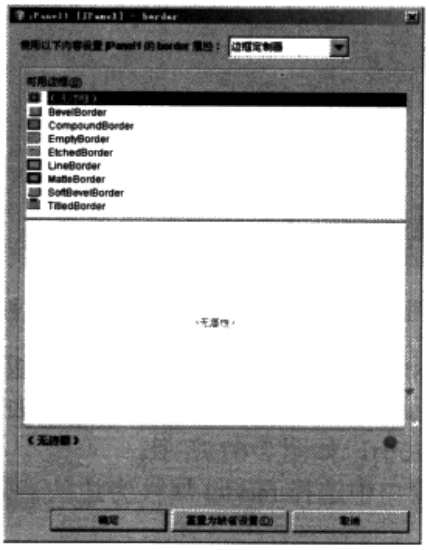


图 2-70 border 对话框

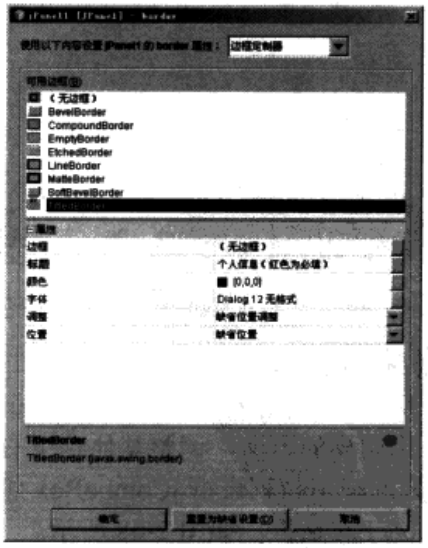


图 2-71 修改边框的标题

❽ 单击“确定”按钮，返回 GUI Builder，此时的界面如图 2-72 所示。

Download at http://www.pin5i.com/

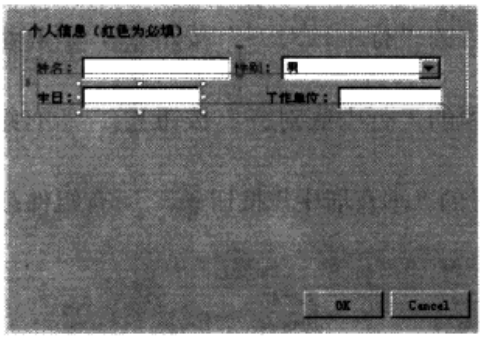


图 2-72 修改完属性的界面

至此，完成了组件属性的修改工作。组件其他属性的修改都可以通过“属性”窗口完成，这里不再一一介绍。

2.5.4 设置组件对齐以及等宽/等高

在编辑界面时，经常需要将组件进行对齐或者设置组件等宽/等高，以获得良好的界面效果。下面仍然以上面的例子为例，介绍组件对齐以及等宽/等高的修改方法。

在“设计”视图中选中左边的两个文本框，然后选择上下文菜单的“相同大小”→“相同宽度”命令，如图 2-73 所示。如果要把几个组件的高度设置为相等，那么这里应该选择“相同高度”命令。

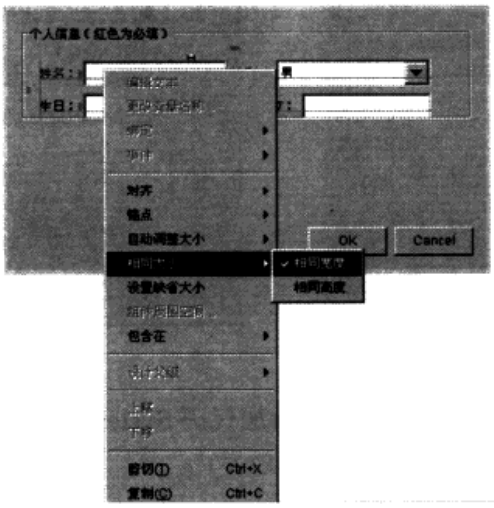


图 2-73 设置等宽

提示：按下 Ctrl 键，用鼠标单击组件可以选中多个组件。用鼠标也可以一次框选多个组件。

选中右边的两个 jLabel 组件，然后选择上下文菜单中的“对齐”→“列左侧”命令，如图 2-74 所示，另外也可以选择工具栏上的“列中左对齐”按钮。其他几个对齐命令的含义如下所示。

- 列左侧：相当于工具栏的“列中左对齐”按钮，所有组件都以列的左边界为基准进行对齐。
- 列右侧：相当于工具栏的“列中右对齐”按钮，所有组件都以列的右边界为基准进行对齐。

Download at http://www.pin5i.com/

- 列中心：相当于工具栏的“水平居中”按钮，所有组件都以列的中心为基准进行对齐。
- 行顶部：相当于工具栏的“行中顶对齐”按钮，所有组件都以行的上边界为基准进行对齐。
- 行底部：相当于工具栏的“行中底对齐”按钮，所有组件都以行的下边界为基准进行对齐。
- 行中心：相当于工具栏的“垂直居中”按钮，所有组件都以行的中心为基准进行对齐。

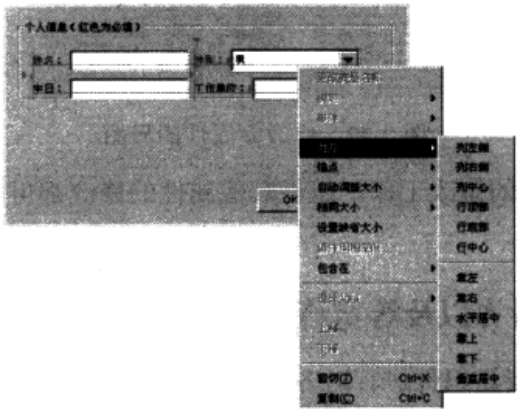


图 2-74　进行对齐操作

对齐后，再调整“工作单位”文本框的大小，使得界面如图 2-75 所示。设计完成后，可以单击“设计”视图工具栏上的“预览设计”按钮，预览设计效果，如图 2-76 所示。

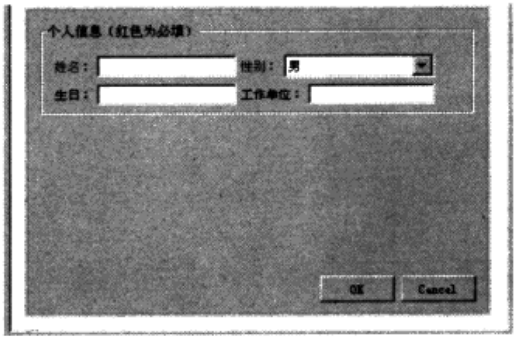


图 2-75　进行对齐和等宽操作后的界面

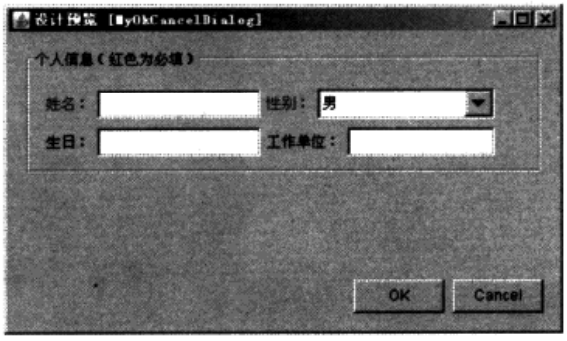


图 2-76　预览设计效果

在运行程序前，需要在 doClose 函数中添加代码 System.exit(0)，这样退出对话框就会退出整个应用程序。doClose 函数的实现代码如下所示：

```
private void doClose(int retStatus) {
    returnStatus = retStatus;
    setVisible(false);
    dispose();
    System.exit(0);
}
```

至此，基本上完成了一个标准应用程序的设计，应用程序的实际运行界面如图 2-77 所示。

2.5.5　布局管理器和 GridBagLayout 布局

NetBeans 提供了一个布局管理器（LayoutManager 接口），通过这个布局管理器可以确定容器内组件的大小和位置，控制在 GUI 窗体中排列可视组件的方式。在容器上单击右键，然后选择弹出菜单中的“设置布局”级联菜单，即可设置/修改容器的布局。

Download at http://www.pin5i.com/

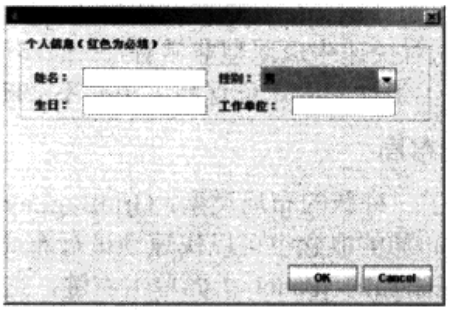


图 2-77 应用程序的实际运行界面

1. 布局概述

默认情况下，GUI Builder 使用的是自由布局（FreeDesign）类型，可以在其中使用可视基准线（自动建议组件的最佳对齐方式和间距）来布置窗体。在工作时，GUI 生成器将设计理念转化为正常工作的 UI，而无须指定布局管理器。由于自由布局类型使用动态布局模型，因此，只要调整窗体大小或切换语言环境，GUI 就会自动调整以适应这些更改，而不会改变组件之间的关系。处理自由布局模型外，NetBeans 还支持如下几种布局模型。

- FlowLayout：FlowLayout 布局如同书页上的字一样在容器中排列组件。它从左到右填充顶行，直到不能容纳任何其他组件为止，然后再以同样的方式继续填充后面的每一行。
- BorderLayout：BorderLayout 布局沿容器的边缘或中心排列组件。使用该布局可以将组件放在以下五个可能的位置：“北”、“南”、“东”、“西”和“中心”，分别对应于容器的顶部、底部、右边缘和左边缘以及内部区域。
- GridLayout：GridLayout 布局将组件放在单元格大小相同的网格中，并将组件从左到右、从上到下添加到网格中。
- GridBagLayout：GridBagLayout 布局是一个功能强大的布局管理器，它使用一组复杂的组件属性（称为“约束”）来精确控制布局的各个方面，甚至在调整容器大小时也是如此。它对于多平台 Java 应用程序特别有用，因为可使用它来创建自由格式的布局，这些布局在所有平台上保持一致的外观。GridBagLayout 将组件放在由行和列组成的网格中，并非所有网格单元的大小都是相同的。此外，组件可以跨越多行或多列，或同时跨越多行和多列。
- CardLayout：CardLayout 布局提供了一种管理占据同一显示区的两个或多个组件的方法。在使用 CardLayout 时，每个组件就像一副纸牌中的一张牌，所有纸牌的大小都相同，且任何时候都只能看到顶部的牌。因为组件占据同一显示空间，所以在设计时必须使用“检查器”窗口选择各个组件。
- BoxLayout：BoxLayout 布局允许垂直或水平（但不能同时在两个方向）排列多个组件。BoxLayout 管理的组件按添加到容器的顺序，从左到右或从上到下排列。添加更多组件时，甚至在调整容器大小时，BoxLayout 中的组件也不能换到第二行或第二列。
- AbsoluteLayout：AbsoluteLayout 布局是一种特殊的 IDE 布局管理器，可使用它将组件准确放在窗体中的所需位置，在 IDE 中任意移动组件以及使用组件选择边框来调整其大小。这对于生成原型特别有用，因为没有任何格式限制，并且不需要输入任何属性设置。



注意：这里建议不要将 AbsoluteLayout 用于生成应用程序，因为当环境发生改变时，组件的固定位置和大小并不会随之改变。另外，AbsoluteLayout 布局不属于标准的 Java 布局，因此如果使用这种布局，在分发应用程序时必须包括 NetBeans IDE 的 AbsoluteLayout 类。可以将这些类部署到<install-dir>/ide6/modules/ext/Absolute Layout.jar 中。

Download at http://www.pin5i.com/

- Null Layout：可使用 Null Layout 布局来设计窗体，而无须使用任何布局管理器。与 AbsoluteLayout 类似，它对快速生成原型非常有用，但建议不要将其用于生产应用程序，因为当环境发生改变时，组件的固定位置和大小并不会随之改变。

2．使用 GridBagLayout 布局

从 NetBeans 5.0 开始加入了一种新的布局类型：GridBagLayout。这种布局为开发人员提供了很大的灵活性，即使不懂得布局的概念也可以快速地进行布局设计。

添加一个 jPanel 组件，在添加的 jPanel 上面单击右键，选择弹出菜单中的“设置布局”→GridBagLayout 命令，设置布局类型为 GridBagLayout 布局。在 jPanel 组件上添加三个 jLabel 组件和三个 jTextField 组件，如图 2-78 所示。

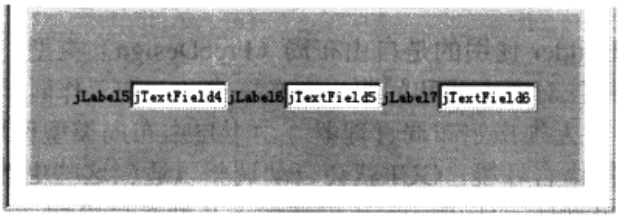


图 2-78　添加界面组件

在 jPanel 上面单击右键，选择弹出菜单中的“定制布局”命令，打开“GridBagLayout 定制器”对话框，如图 2-79 所示。

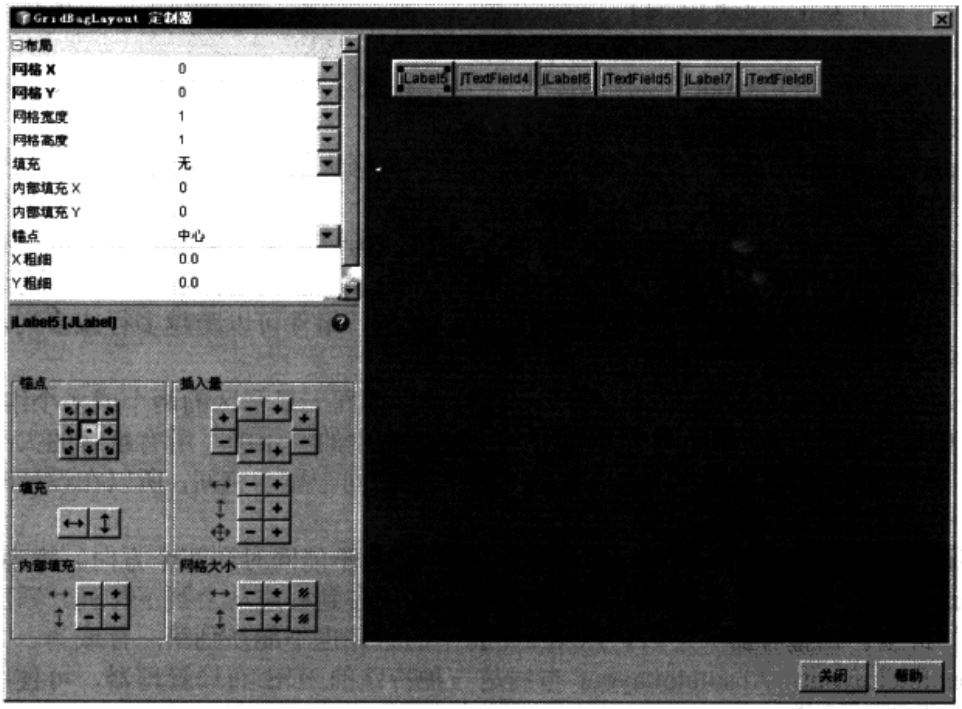


图 2-79 “GridBagLayout 定制器”对话框

“GridBagLayout 定制器”对话框可在 GridBagLayout 中对组件的位置和约束进行可视化调整。它包含 GridBag 约束的属性表单、用于调整约束的按钮以及对组件布局的简短描述。拖动右窗格中的组件，根据需要对其重新定位。拖动组件时，其“网格 X”和“网格 Y”属性也随之改变，以反映新位置，如图 2-80 所示。

Download at http://www.pin5i.com/

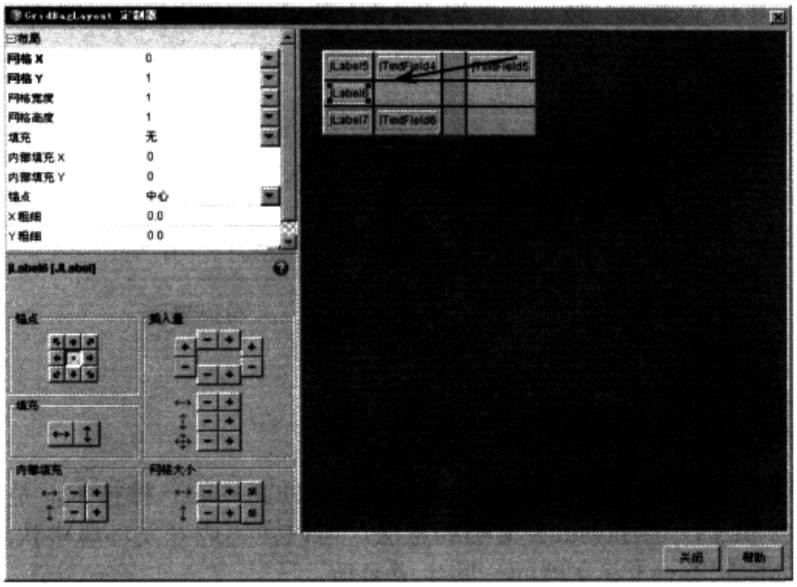


图 2-80 调整组件

在完成组件的大概布局后，选择一个组件，并在左窗格中根据需要调整其约束。可以调整以下约束。

- “网格 X”约束和“网格 Y”约束：通过设置组件的网格 X 和网格 Y 的位置来微调其水平和垂直位置。
- “网格宽度”和“网格高度”约束：可使用“网格大小”按钮调整“网格宽度”约束和“网格高度”约束。单击最右侧带有黄色阴影的按钮，将该值设置为“余量”。也可以直接设置“网格宽度”约束和“网格高度”约束值，以指定各个方向上为组件分配的网格位置数。可以选择以下选项之一。
 - 整数值——组件使用的单元数（而非像素数）。
 - 余量——使用所有剩余的水平或垂直空间，使组件成为所在行或列的最后一项。
 - 相对——指定组件紧挨所在行或列中的最后一项。
- “填充”约束：可使用“填充”按钮调整或者直接设置约束值。“填充”约束用于设置组件是使用为其分配的所有垂直空间、水平空间，还是两部分都使用。对于分配给组件且该组件未填充的任何空间，都将在右窗格中使用蓝色进行标记。
- “内部填充 X”约束和“内部填充 Y”约束：可以使用内部填充设置来增加组件的水平和垂直尺寸。可通过直接输入约束值或使用“内部填充”按钮来调整这些值。
- “锚点”约束：可以使用“锚点”约束将组件置于为其分配的空间内九个位置中的任意一处（“中心”、“北”和“西北”等）。如果组件没有剩余的空闲空间，则此设置不起作用。
- “X 权重”约束和“Y 权重”约束：调整权重设置来确定在调整容器窗口大小时应为组件分配的空间（相对于所在行或列中的其他组件）。权重值通常在 0 到 1 之间。调整窗口大小时，权重值较大的组件在其行或列中获得的空间更多。权重值为 0 的组件始终保持该方向的首选大小。如果行或列中所有组件的权重都为 0，则任何额外的空间都将超出行或列的外边界，且组件大小将保持不变。
- “插入量”约束：“插入量”约束用于确定组件四侧的最小外部空间。可手动输入数值，也可以使用“插入量”按钮调整。“插入量”按钮分成四组，以便单独调整各侧的插入量、同时调整左右两侧的插入量或同时调整顶部和底部的插入量以及同时调整四侧的插入量。更改插入量时，将在右窗格中显示以黄色背景标记的插入区域。

Download at http://www.pin5i.com/

修改完成后，单击“关闭”按钮，此时在 GUI Builder 中会显示出编辑的界面布局效果，如图 2-81 所示。

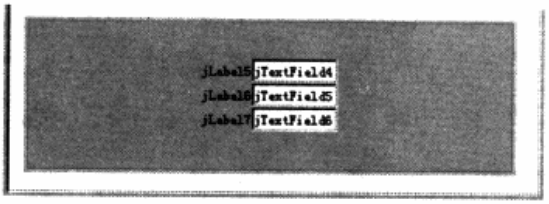


图 2-81 定制后的 GridBagLayout 布局界面

2.6 使用 Javadoc

Javadoc 是根据在源代码文件中的注释信息产生帮助文档的一种技术，是 Java 特有的一种技术。NetBeans 提供了产生 Javadoc 文档的方法，可以在集成开发环境中快速地产生 Javadoc 文档，并且还可以控制 Javadoc 文档的一些生成参数。

2.6.1 设置 Javadoc 生成参数

对于每个项目都可以设置项目的属性来控制生成 Javadoc 文档的内容。在“项目”窗口中右键单击项目，选择上下文菜单中的“属性”命令打开“项目属性”对话框。在对话框左边的“类别”列表中选择“生成”→“生成文档”节点，显示生成 Javadoc 文档的配置参数，如图 2-82 所示。

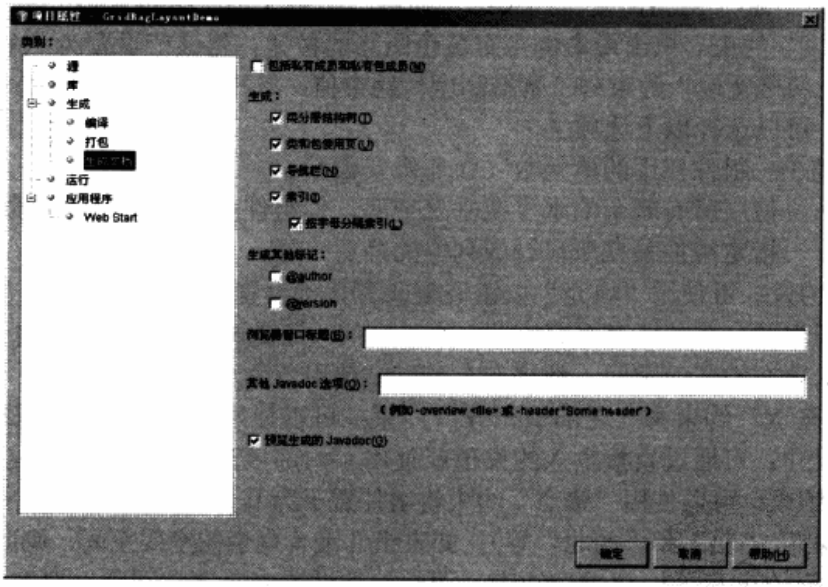


图 2-82 生成 Javadoc 文档的配置参数

如果要在 Javadoc 文档中包含私有成员和私有包成员的说明，那么应该选中“包括私有成员和私有包成员”复选框。

如果要在 Javadoc 文档中包含作者信息，那么要选中“@author”复选框；如果要在 Javadoc 文档中包含版本信息，那么要选中“@version”复选框。

Javadoc 文档浏览窗口的标题在“浏览器窗口标题”文本框中输入。其他 Javadoc 选项则在“其他 Javadoc 选项”文本框中输入。

Download at http://www.pin5i.com/

2.6.2 生成 Javadoc

假设主项目的名字为“Abc”，那么选择“生成”→“为 Abc 生成 Javadoc”命令可以为该主项目生成 Javadoc 文档，生成的 Javadoc 文档如图 2-83 所示。

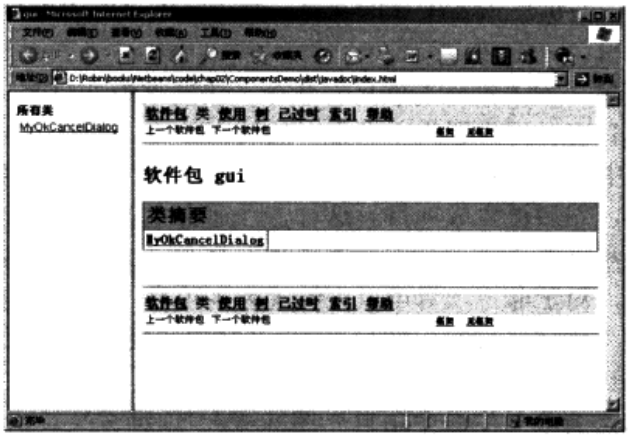


图 2-83 生成的 Javadoc 文档

2.7 程序的打包与发布

在程序编写完成后，需要打包进行发布，把编译后的.class 文件打包为 JAR 文件。本节将讨论程序打包的具体设置以及打包方法。

2.7.1 设置包装选项

对于每个项目都可以设置项目的属性来控制打包参数。在“项目”窗口中右键单击项目，选择上下文菜单中的“属性”命令打开“项目属性”对话框。在对话框左边的“类别”列表中选择“生成”→“打包”节点，显示打包的具体配置参数，如图 2-84 所示。

在“JAR 文件”文本框中给出了生成的 JAR 文件的具体路径。“从 JAR 文件中排除”文本框中给出了打包时排除的文件。这里*.java 文件为源代码文件，因此不应该包含在发布包中；*.form 文件为 NetBeans 的窗体文件，也不应该包含在发布包中。如果希望压缩生成 JAR 文件，则应该选中“压缩 JAR 文件”复选框。

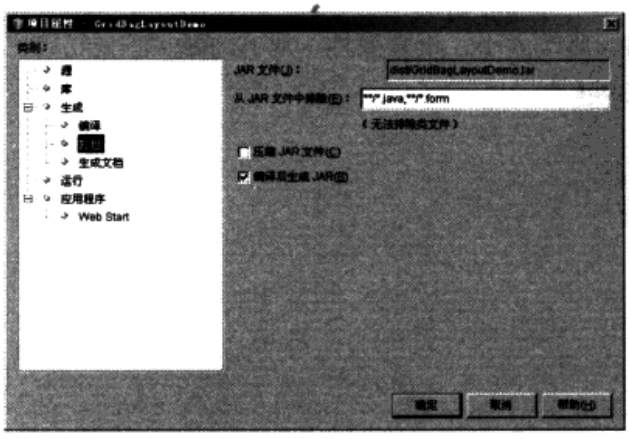


图 2-84 设置包装选项

Download at http://www.pin5i.com/

2.7.2 项目的生成

NetBeans 的项目基础结构直接建立在 Apache Ant 之上，因此可以使用 Ant 脚本来运行 Java 应用程序。下面几种方式都可以用来生成项目。

- 在“项目”窗口中右键单击要生成的项目，然后单击“生成项目”命令。这种方法适合主项目和非主项目。
- 选择“生成”菜单下的“生成主项目”命令（快捷键为 F11）。这种方法只适用于主项目，对于非主项目可以先设置为主项目，然后再使用这种方法。
- 在“项目”窗口中右键单击要生成的项目，然后单击“清除并生成项目”命令。此时会先执行清除操作，然后再生成项目。这种方法适合主项目和非主项目。
- 选择“生成”菜单下的“清除并生成主项目”命令（快捷键为 Shift + F11）。这种方法只适用于主项目，对于非主项目可以先设置为主项目，再使用这种方法。
- 单击工具栏上的“生成主项目”图标 。这种方法只适用于主项目，对于非主项目可以先设置为主项目，然后再使用这种方法。
- 单击工具栏上的“清除并生成主项目”图标 。这种方法只适用于主项目，对于非主项目可以先设置为主项目，然后再使用这种方法。



说明：如果只想清除项目，可以在“项目”窗口中右键单击要清除的项目，然后选择弹出菜单中的“清除项目”命令即可。

2.8 小结

本章主要讲解使用 NetBeans 开发 Java 标准项目的方法，其中重点介绍了如何通过模板生成标准的应用程序和对话框以及如何添加和编辑界面组件。通过本章的学习，读者应该掌握以下内容：

- 项目的打开和关闭方法
- 如何设置主项目
- 如何从 Eclipse 和 Jbuilder 中导入项目
- 项目的编译和生成
- 如何通过模板生成标准的应用程序
- 如何通过模板创建不同的对话框
- 界面组件的添加、修改大小、设置属性、对齐方法
- 使用 GridBagLayout 布局

Download at http://www.pin5i.com/

第3章

在 NetBeans 中调试应用程序

在开发程序的时候，可能会出现各种错误，虽然编译器会帮助开发人员找到一些语法错误，但是对于一些逻辑错误或者算法上的错误，就只能利用 NetBeans 的调试技术进行分析和解决了。本章将讨论 NetBeans 如何调试应用程序、进行单步跟踪、查看本地变量等内容。

本章主要内容

- 设置和管理断点
- 如何执行单步跟踪
- “监视”窗口的使用
- 局部变量窗口
- 如何在调试时修改代码

本章亮点

- NetBeans 各个类型断点的设置方法
- 各个调试窗口的使用
- 如何计算表达式

3.1 在 NetBeans 中调试应用程序

调试是检查应用程序是否存在错误的过程。可以通过在代码中设置断点和监视，然后在调试器中运行代码来完成调试过程。NetBeans 集成开发环境使用 Sun Microsystems JPDA 调试器来调试程序。选择“运行”→“步入”命令或者按下快捷键 F7，就可以进入调试会话界面。此时，程序会停止在代码的入口处，如图 3-1 所示。

另外，也可以选择“运行”→“运行至光标”命令或者按下快捷键 F4，可以将程序运行到源代码编辑器中光标所在的位置并暂停程序。

在启动调试会话时，所有相关的调试器窗口都将自动出现在屏幕的底部。在“输出”窗口中会添加一个“调试器控制台”标签，用于显示来自当前会话调试器的消息，并包括指向源代码的超级链接。在“输出”窗口中可以使用如下的快捷菜单命令。

- “复制”命令：将选定文本复制到剪贴板。
- “查找”命令：在窗口中查找字符串。
- “查找下一个”命令：查找在“查找”中指定的搜索项的下一个出现位置。
- “自动换行”命令：如果选中该选项，文本将在窗口宽度位置换行。如果未选中，则必须扩展窗口以查看比窗口更宽的文本。
- “另存为”命令：提示您输入可以保存窗口内容的文件。
- “清除”命令：清除选定标签化窗口的内容。
- “关闭”命令：完全关闭“输出”窗口。

使用以下任何一种方式都可以关闭调试会话：

Download at http://www.pin5i.com/

- 选择“运行”→“完成调试器会话”命令；
- 按下 Shift + F5 组合键；
- 单击“调试”工具栏中的“完成调试器会话”按钮 。

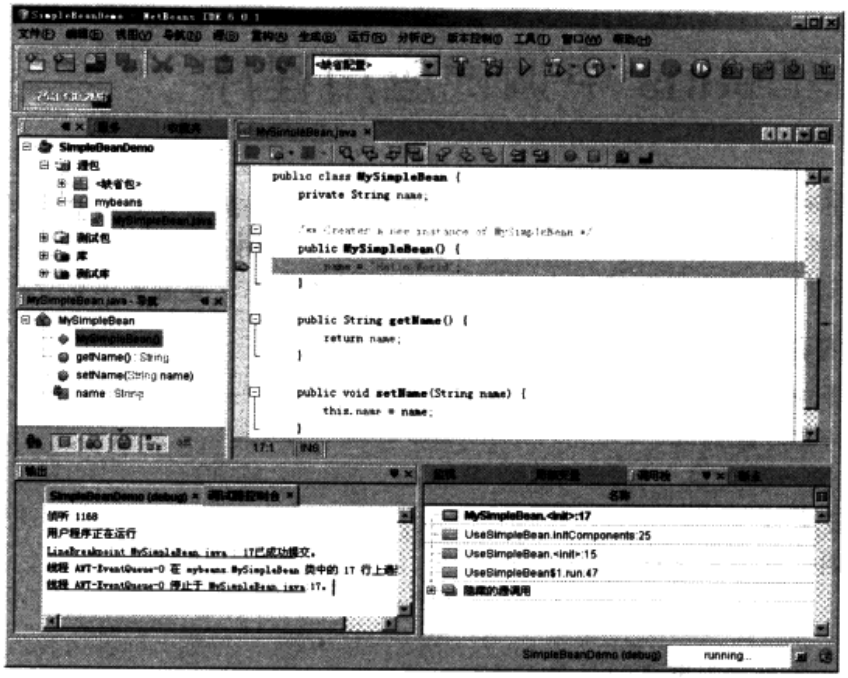


图 3-1 调试会话界面

3.2 设置和管理断点

断点是源代码中的标志，它会通知调试器停止执行程序。当程序在断点处停止时，开发人员可以执行诸如检查变量值和单步执行程序等操作。

3.2.1 断点类型

NetBeans 中支持以下几种断点类型。

- 行断点：可以为某一个代码行设置断点，这样当运行到该行时就会中断程序的执行。
- 类断点：可以在将类装入虚拟机和/或从虚拟机中卸载类时中断程序的执行。
- 异常断点：可以在捕捉到特定异常、未在源代码中处理特定异常或遇到任何异常（无论程序是否处理错误）时中断程序的执行。
- 变量断点：可以在访问（例如，将变量作为参数来调用方法）或修改特定类和字段中的变量时中断程序的执行。
- 方法断点：每次执行该方法时都将中断程序的执行。
- 线程断点：可以在线程开始和/或停止时中断程序的执行。

说明：除了行断点之外，所有 Java 断点都是全局定义的，因此它们会影响所有集成开发环境项目（这些项目包含设置了断点的源）。例如，如果在某个项目中的 com.mycompany.MyOkCancelDialog 上设置了类断点，则在其他项目（包含该类）的调试会话期间，集成开发环境每次遇到该类时都会停止执行。

Download at http://www.pin5i.com/

如果已经设置了断点，对于主项目，可以选择“运行”→“调试主项目”命令或者按下快捷键 F5 进入调试会话，程序将中断在遇到的第一个断点的位置。

对于非主项目，可以在“项目”窗口中右键单击该项目，然后选择弹出菜单中的“调试项目”命令进入调试会话，程序将中断在遇到的第一个断点的位置。

对于某个文件，可以在“项目”窗口中选择任何可运行的文件，然后选择“运行”→“运行文件”→“调试<选中的文件名>”命令或者按下 Ctrl + Shift + F5 组合键进入调试会话，程序将中断在遇到的第一个断点的位置。



3.2.2 添加行断点

行断点的设置比较简单。在源代码编辑器中单击要设置断点行的左边距，或者按 Ctrl + F8 组合键。添加断点后，设置断点的行会以红色突出显示，并在左边距中显示一个的红色的方框，如图 3-2 所示。再次单击该行的左边或者按 Ctrl + F8 组合键将取消该断点。

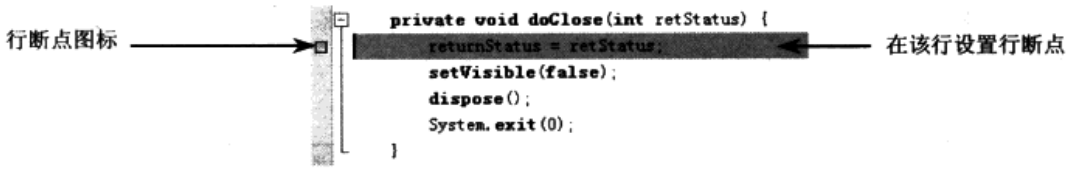


图 3-2 设置行断点



3.2.3 添加类断点

如果要设置除了行断点以外的其他类型的断点，可以在源代码编辑器中选择要设置断点的代码元素。例如，如果要在类 MyOkCancelDialog 上设置类断点，请在类声明中选择该类名。然后，选择“运行”→“新建断点”命令或者按下 Ctrl＋Shift＋F8 组合键，打开“新建断点”对话框，如图 3-3 所示。

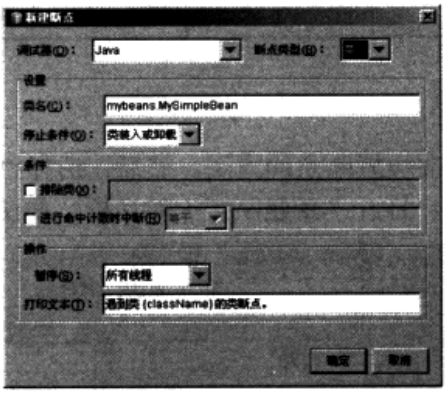


图 3-3 在“新建断点”对话框中添加类断点

添加类断点的各个参数说明如下：

- “调试器”下拉列表框：选择调试器的类型，如果是调试 Java 程序，应该选择 Java 调试器。
- “断点类型”下拉列表框：说明当前的断点类型为类断点。如果要设置为其他的断点类型，可以通过下拉列表框进行选择。
- “类名”文本框：用于指定为哪个类（包含包名的全路径）设置类断点。
- “停止条件”下拉列表框：用于指定触发断点的操作，可以有如下的选项。
 - 类装入：将类加载到虚拟机时触发断点的操作。

Download at http://www.pin5i.com/

 - 类卸载：从虚拟机卸载类时触发断点的操作。
 - 类装入和卸载：将类加载到虚拟机时或者从虚拟机卸载类时触发断点的操作。
- “排除类”复选框：选中该复选框后，可以在后面的文本框中输入要排除的类。被排除的类中的断点将会被排除，即不会在这些断点处暂停执行。
- “进行命中计数时中断”复选框：选中该复选框可以在后面指定断点触发的频率和时间。例如调试一个循环语句时，如果不希望每个循环都触发断点，可以使用该选项，设置循环执行 100 次以后才会触发断点。
- “暂停”下拉列表框：指定到达断点时调试的程序中要暂停的线程，可以有如下的选项。
 - 无线程（继续）：将继续执行。
 - 所有线程：所用线程都将暂停执行。
 - 断点线程：到达断点时的当前线程暂停执行。
- “打印文本”文本框：将在“输出”窗口的“调试器控制台”视图中打印指定的消息。这里将打印类断点的类名。打印消息中可以使用特殊的标记，表示特殊的含义。标记的定义以及使用方法如表 3-1 所示。其中 ✓ 或文本表示可将标记用于该断点类型。✖ 表示该标记不能用于该断点类型。

表 3-1 断点打印消息中的特殊标记

标 记	描 述	断点类型					
		行	方法	异常	线程	类	变量
{className}	打印类的名字	✓	✓	✓	✖	✓	✓
{lineNumber}	打印行号	✓	✓	✓	✖	✓	✓
{methodName}	打印方法的名字	✓	✓	✓	✖	✓	✓
{threadName}	打印线程的名字	✓	✓	✓	✓	✓	✓
{variableValue}	打印变量值	✖	✖	异常值	✖	✖	变量值
{variableType}	打印变量类型	✖	✖	异常名	✖	✖	变量类型
{action}	打印动作	✖	✖	✖	✖	准备或卸载	修改或访问
{$"expression"}	打印表达式	✓	✓	✖	✖	✖	✓



3.2.4 添加方法断点

在源代码编辑器中，选择要设置断点的方法，然后，选择“运行”→“新建断点”命令或者按下 Ctrl＋Shift＋F8 组合键，打开“新建断点”对话框，如图 3-4 所示。

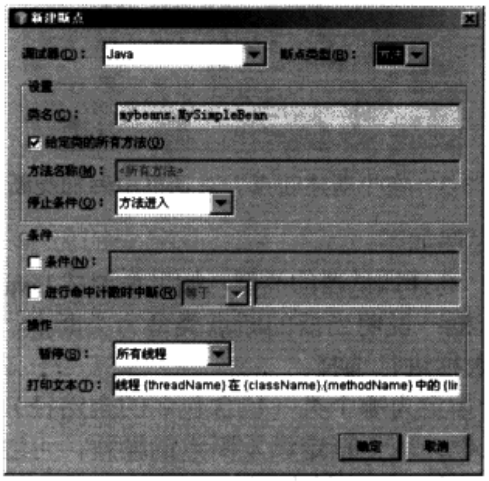


图 3-4 在“新建断点”对话框中添加方法断点

Download at http://www.pin5i.com/

添加方法断点的参数和类断点参数基本一致。如果选中“给定类的所有方法”复选框，则会将断点应用于指定类的所有方法。如果取消该复选框，则要在“方法名称”文本框中输入断点的方法名称。

3.2.5 添加异常断点

在如图 3-3 所示的“新建断点”对话框中，从“断点类型”下拉列表中选择“异常”选项，创建一个异常断点，如图 3-5 所示。

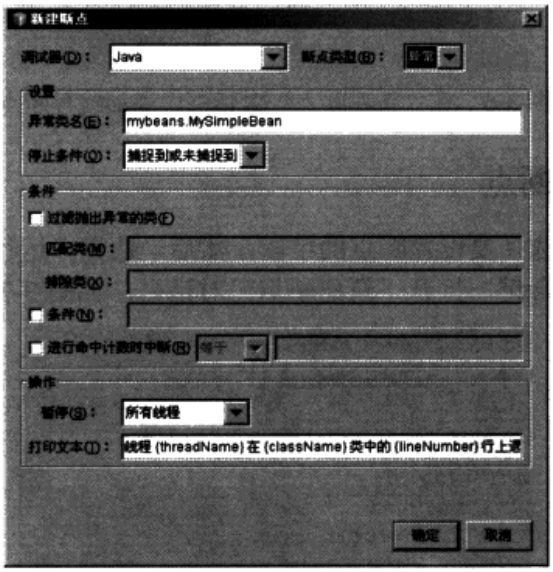


图 3-5 在“新建断点”对话框中添加异常断点

其中，“操作”复选框中各个参数的含义请参考第 3.2.3 节关于添加类断点的参数说明。其他添加异常断点的参数说明如下。

- “异常类名”文本框：选择或者输入要设置异常断点的异常类或异常超类的名称。
- “停止条件”下拉列表框：指定触发断点的操作，可以有如下的选项。
 - 捕捉到：仅在捕捉到异常时中断程序的执行。
 - 未捕捉到：仅在未捕捉到异常时中断程序的执行。
 - 捕捉到或未捕捉到：所有异常处都中断程序的执行。
- “过滤抛出异常的类”复选框：选中该复选框，可以过滤触发异常断点的类。此时，可以在下面的“匹配类”文本框中输入允许触发异常断点的类，只有输入的这些类中的异常断点才会被触发。当然，也可以在“排除类”文本框中输入希望排除的类名字，这些被排除的类中的异常断点将不会被触发。

3.2.6 添加线程断点

在如图 3-3 所示的“新建断点”对话框中，从“断点类型”下拉列表中选择“线程”选项，创建一个线程断点，如图 3-6 所示。

在“停止条件”下拉列表框用于指定触发断点的操作，可以选择如下的选项。

- 线程开始：在线程开始时触发断点。
- 线程停止：在线程停止时触发断点。
- 线程开始或停止：既在线程开始时触发断点，又在线程停止时触发断点。

Download at http://www.pin5i.com/

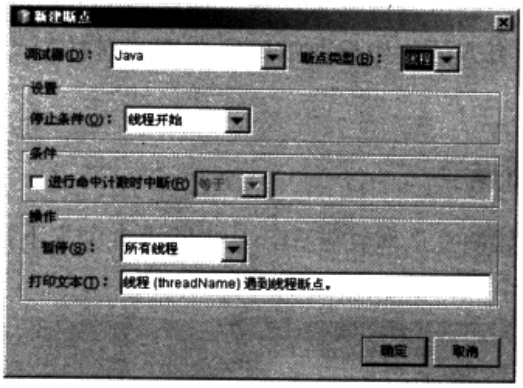


图 3-6 在“新建断点”对话框中添加线程断点

3.2.7 添加字段断点

在源代码编辑器中，选择要设置断点的字段，然后，选择“运行”→“新建断点”命令或者按下 Ctrl＋Shift＋F8 组合键，打开“新建断点”对话框，如图 3-7 所示。

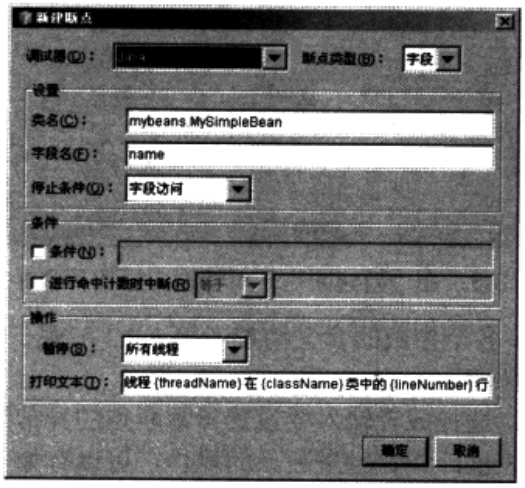


图 3-7 在“新建断点”对话框中添加**字段**断点

其中，“操作”分组框中各个参数的含义请参考第 3.2.3 节关于添加类断点的参数说明。其他添加异常断点的参数说明如下。

- “类名”文本框：包含字段（该字段要被设置为字段断点）的 Java 类文件全路径名。也可使用星号作为类名。例如，如果将 java.lang 指定为包名，并将*作为类名，则断点将会应用于 java.lang 包的所有类。
- “字段名”文本框：要设置字段断点的字段的名字。
- “停止条件”下拉列表框：指定触发断点的时间，可以有如下的选项。
 - 字段访问：查询字段时，将中断程序的运行。
 - 字段修改：修改字段时，将中断程序的运行。
 - 字段访问或修改：在查询或者修改字段时，都会中断程序的运行。

3.2.8 断点的管理

NetBeans 集成开发环境提供了“断点”窗口，以便于对断点进行管理。选择“窗口”→“调试”→“断点”命令或者按下 Alt + Shift + 5 组合键，打开“断点”窗口，如图 3-8 所示。

Download at http://www.pin5i.com/

在名称列表中，列出了所用的断点。选中某个断点，然后单击右边的启用列，可以禁止或者启用某个断点。另外，在“断点”窗口中右键单击某个断点，然后选择弹出菜单中的“启用”或“禁用”命令也可以启用或者禁止某个断点。在“断点”窗口中单击右键，然后选择弹出菜单的“全部启用”或“全部禁用”命令则可以启用或者禁止全部的断点。

“断点”窗口弹出菜单中的“定制”命令可以打开“定制断点”对话框，对断点进行修改。“定制断点”对话框和“新建断点”对话框基本一样，请读者参考“新建断点”对话框的使用方法。

“断点”窗口弹出菜单中的“设置组名称”命令可以打开“设置断点组的名称”对话框，如图 3-9 所示。

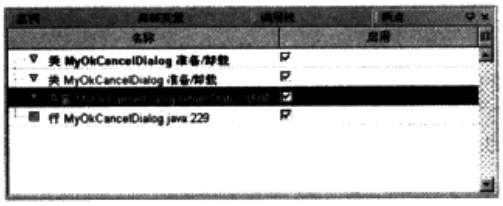


图 3-8 “断点”窗口

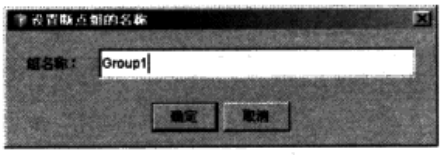


图 3-9 “设置断点组的名称”对话框

在“组名称”文本框中输入任何的组名字，单击“确定”按钮就可以把一个断点分到这个组中。如果在集成开发环境中设置了很多断点，则可以将这些断点归到不同的组，这样就可以将它们作为一个单元进行启用、禁用和删除操作。图 3-10 是对断点进行了简单分组后的显示界面。

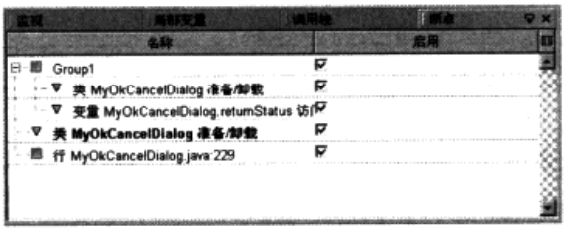


图 3-10 对断点进行分组后的“断点”窗口

3.3 单步执行

实际开发中，断点一般都要和单步执行配合进行调试。NetBeans 提供了如下的几种单步执行方法。

- 越过：单步执行一行源代码，如果源代码行包含调用，则执行整个例程而不单步执行单个指令。
- 步入：单步执行一行源代码。如果源代码行包含调用，则进入该调用例程，在执行例程的第一条语句之前停止。如果用“步入”方法启动调试会话（即选择“运行”→“步入”命令），程序会在 main 入口方法的第一行停止执行。
- 步出：单步执行一行源代码。如果源代码行是某个例程的一部分，则会执行该例程的其余各行，然后停止在调用该例程的行。

在调试过程中，这三种方法需要互相配合使用。例如，图 3-11 所示的调试会话中，执行越过命令，代码将停止在第 27 行。注意，如果此时仍然执行越过操作，则会执行到第 28 行。这里需要查看 getReturnStatus 方法内部的执行情况，因此执行步入命令，代码将停止在第 35 行。在检查完 getReturnStatus 方法后，执行步出命令，程序将会返回第 27 行。

Download at http://www.pin5i.com/


```
25    private void save() {
26        String str = jTextField1.getText();
27        int status = getReturnStatus();
28        if (status == RET_OK) {
29            jTextField4.setText("OK");
30        }
31        jTextField3.setText("ABC");
32    }
33    /** @return the return status of this dialog - one of RET_OK or RET_CANCEL */
34    public int getReturnStatus() {
35        return returnStatus;
36    }
```

图 3-11 单步执行示例

表 3-2 给出了不同单步调试方法的执行方法。

表 3-2 单步调试的执行方法

单步调试方法	执行方法
越过	选择“运行”→“越过”命令
	按下快捷键 F8
	选择“调试”工具栏的“越过”按钮
步入	选择“运行”→“步入”命令
	按下快捷键 F7
	选择“调试”工具栏的“步入”按钮
步出	选择“运行”→“步出”命令
	按下快捷键 Ctrl + F7
	选择“调试”工具栏的“步出”按钮



说明：在调试会话中，如果选择“运行”→“继续”命令（快捷键 Ctrl + F5），或者选择“调试”工具栏中的“继续”按钮，则会继续执行代码一直到程序结束或者遇到下一个断点。



3.4 查看变量和表达式的值

在进行调试的时候，有时候需要知道变量和表达式的值，以确认执行得是否正确。NetBeans 的调试会话支持查看变量和表达式的值。



3.4.1 在代码编辑器中直接查看

调试会话中，在程序中断执行后，如果把鼠标放在某个变量的上面，就可以显示该变量的值，如图 3-12 所示。

```
227
228    private void doClose(in retStatus = (int) 0
229        returnStatus = retStatus;
230        setVisible(false);
231        dispose();
232        System.exit(0);
233    }
```

图 3-12 查看变量的值

Download at http://www.pin5i.com/

如果要查看某个表达式的值，则可以选中这个表达式，然后把鼠标放在选中的表达式上。此时会显示表达式以及表达式的值，如图 3-13 所示。

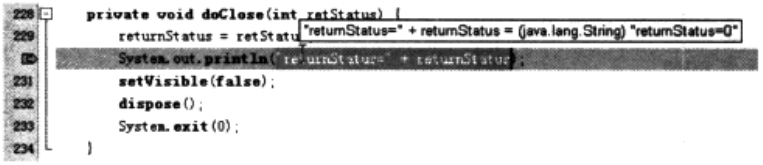


图 3-13 查看表达式的值

3.4.2 使用“监视”窗口查看

NetBeans 还支持创建监视，通过监视在程序执行期间跟踪变量或表达式值的变化。创建监视通过“新建监视”对话框实现，如图 3-14 所示。下面的三种方式都可以打开“新建监视”对话框：

- 在源代码编辑器中选择变量或表达式，单击鼠标右键，然后选择弹出菜单中的“新建监视”命令；
- 按下 Ctrl + Shift + F7 组合键；
- 选择“运行”→“新建监视”命令。

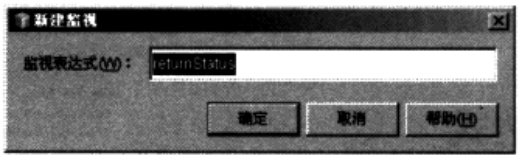


图 3-14 “新建监视”对话框

在“监视表达式”文本框中输入要被监视的表达式，然后单击“确定”按钮，表达式将被添加到“监视”窗口，如图 3-15 所示。



图 3-15 “监视”窗口

“监视”窗口列出了变量和表达式的全部监视。默认情况下，“监视”窗口将在每次启动调试会话时自动打开。也可以通过选择“窗口”→“调试”→“监视”命令来打开“监视”窗口。

在调试会话时，“监视”窗口中的变量和表达式的值会随着代码的执行而改变。当更改当前上下文时，“监视”窗口将进行更新，以显示对应于该上下文的监视值。

“监视”窗口的上下文菜单中，“删除”命令用于从“监视”窗口中删除选定的监视；“全部删除”命令用于从“监视”窗口中删除所有的监视；“属性”命令会打开一个对话框，修改监视表达式。

3.4.3 使用“局部变量”窗口查看

NetBeans 的“局部变量”窗口列出了当前调用中的局部变量，如图 3-16 所示。默认情况下，“局部变量”窗口将在每次启动调试会话时自动打开。也可以通过选择“窗口”→“调试”→“局部变量”命令来打开“局部变量”窗口。

Download at http://www.pin5i.com/

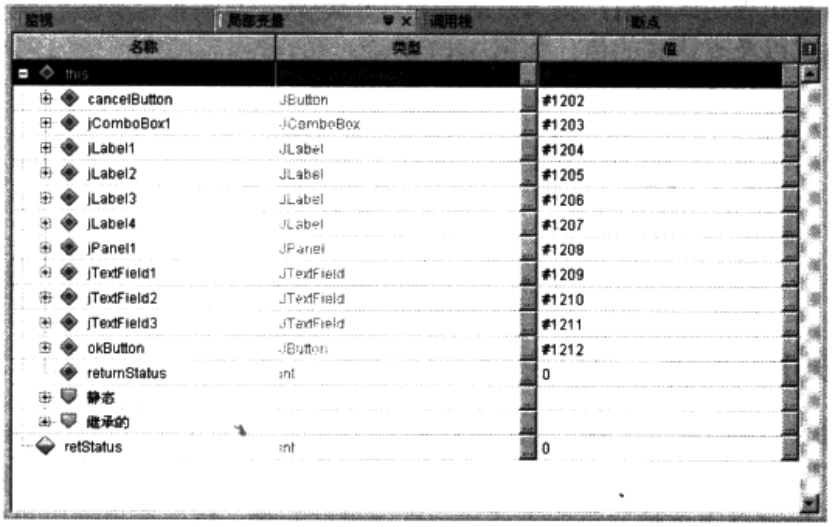


图 3-16 “局部变量”窗口

“局部变量”窗口中列出了每个局部变量的名称、类型和值。可以单击名称左边的控件展开或折叠变量。如果对象类型以带下画线的蓝色文本显示，则单击文本将跳转至源代码中的对象类型。单击“值”单元格可以直接在“局部变量”窗口中编辑值。

3.5 计算表达式

在调试会话中，还可以计算一些代码中不存在的表达式。NetBeans 会根据当前上下文会话中相关变量的值计算出表达式的值。当调试会话上下文发生变化后，表达式的值也会随着变化，以反映最新的值。

在调试会话中，选择“运行”→“计算表达式”命令或者按下 Ctrl + F9 组合键，打开“计算表达式的值”对话框，如图 3-17 所示。这里输入表达式“returnStatus==0”，然后单击“求值”按钮。此时会在“结果”列表中显示计算的结果。单击“监视”按钮则会在“监视”窗口中添加表达式的监视。

“计算表达式的值”对话框可以计算任何表达式的值，以查看当前方法的局部变量、类变量以及方法调用的值。如果计算的方法会更改全局变量，在调试会话中也会更改该全局变量的值。

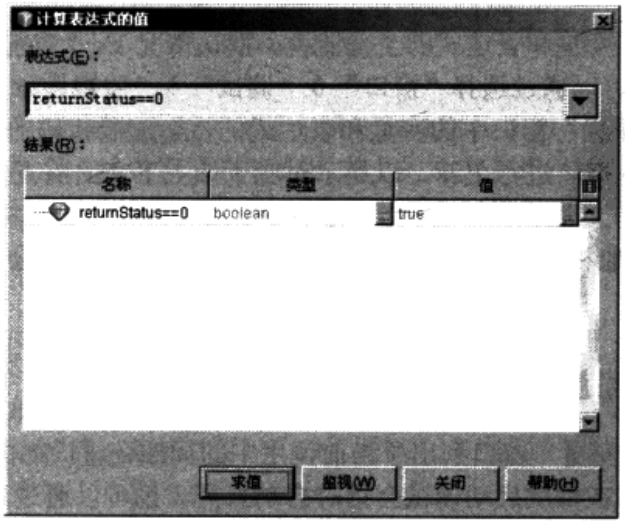


图 3-17 “计算表达式的值”对话框

Download at http://www.pin5i.com/

3.6 在调试模式修改代码

NetBeans 集成开发环境中可以支持在会话模式修改代码，并且在不退出调试会话的情况下进行编译并重新装载更新后的代码。在调试会话时，添加如图 3-18 所示的第 230 行代码，然后选择“运行”→“应用代码更改”命令。

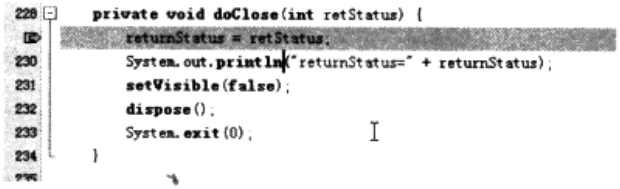


图 3-18 在调试会话中修改代码

NetBeans 会重新编译代码，并装载更新后的代码。在“输出”窗口中将添加一个新的标签，并显示代码修改的过程，如图 3-19 所示。

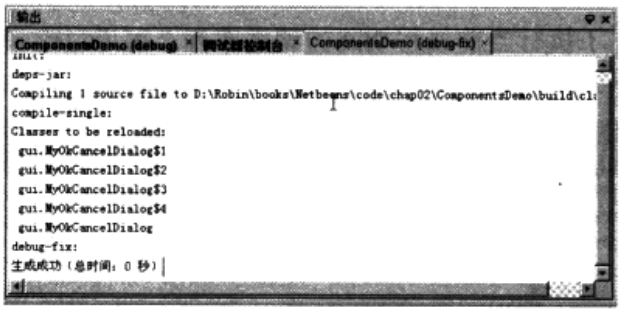


图 3-19 “输出”窗口中的代码修改信息

3.7 NetBeans 的调试窗口

在前面几节中已经介绍了一个 NetBeans 调试窗口的使用方法，下面介绍其他几个调试窗口的使用，包括“类”窗口、“调用栈”窗口、“线程”窗口、“会话”窗口和“源”窗口。

3.7.1 “装入的类”窗口

“装入的类”窗口显示已装入虚拟机中所有类的分层结构，此虚拟机承载当前调试的程序。“类”窗口的内容取决于当前上下文。更改当前会话时，“类”窗口将更新以显示该会话的类。可通过选择“窗口”→“调试”→“装入的类”命令来打开“装入的类”窗口，如图 3-20 所示。

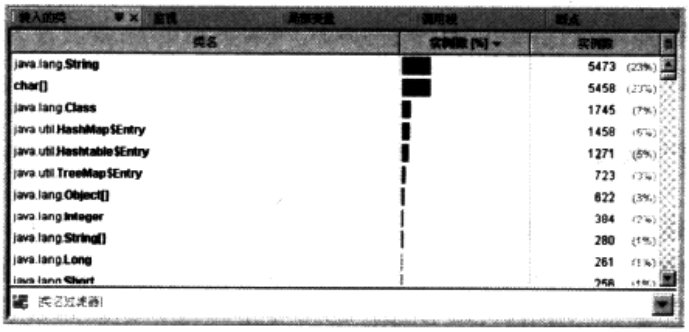


图 3-20 “装入的类”窗口

Download at http://www.pin5i.com/

在“装入的类”窗口中，双击某个类可以打开实例窗口，如图 3-21 所示。在这个窗口中可以查看类的字段、引用实例等内容。

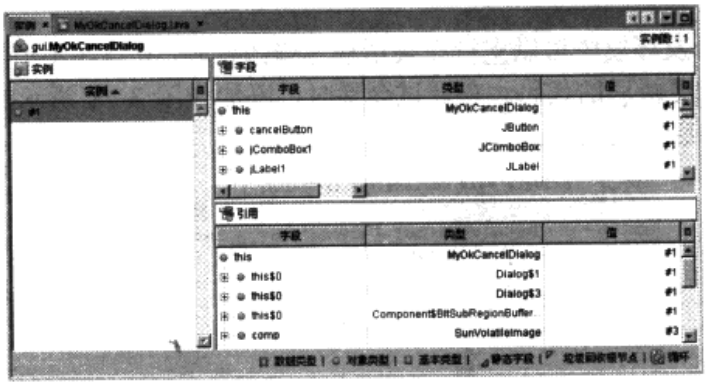


图 3-21 类的实例窗口

3.7.2 “调用栈”窗口

“调用栈”窗口列出了执行当前线程期间的调用序列。可通过选择“窗口”→“调试”→“调用栈”命令或者按下 Alt + Shift + 3 组合键打开“调用栈”窗口，如图 3-22 所示。

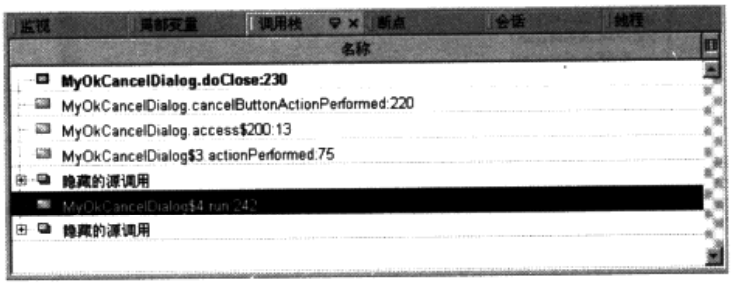


图 3-22 “调用栈”窗口

“调用栈”窗口中，每个调用提供的信息包括：调用名称和调用当前执行语句的行号。如果文件名和行号显示为带下画线的蓝色文本，则表示文本将链接到该调用的源代码。

标有 图标的线程是当前线程最近执行的调用。在“调用栈”窗口中双击某个调用，或者选择某个调用上下文菜单的“激活”命令，会激活该调用，同时调用的图标会变为 。

选择“运行”→“栈”→“弹出最顶层调用”命令，将调用从调用栈中删除。程序计数器被移动到执行删除调用指令的前一行。如果源代码可用，源代码编辑器的焦点将置于该行。继续执行程序时，会重复该调用。

在“调用栈”窗口中，右键单击想要保留在调用栈顶层的调用，然后选择弹出菜单中的“弹出到此处”命令，将会从调用栈中删除选定调用之上的所有调用。程序计数器被移动到执行删除调用指令的前一行。如果源代码可用，则将源代码编辑器的焦点设置为该行。继续执行程序时，会重复该调用。

3.7.3 “线程”窗口

在当前会话中创建的所有线程都将显示在“线程”窗口中。对于 Java 应用程序来说，此列表还包括应用程序中的线程组。选择“窗口”→“调试”→“线程”命令或者按下 Alt + Shift +7 组合键，打开“线程”窗口，如图 3-23 所示。

Download at http://www.pin5i.com/

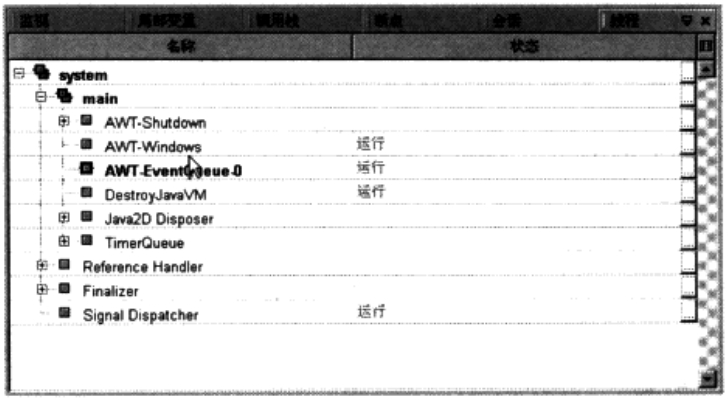


图 3-23 “线程”窗口

在任何给定时间，都只有一个线程是当前线程，当前线程的图标为 。默认情况下，当前线程是当前会话内调试器从中获得控制的线程。在“线程”窗口中，双击任何线程，或者选择线程上下文菜单的“激活”命令，都可以将线程设置为当前线程。

如果需要暂停某一线程，可以在“线程”窗口中右键单击该线程，然后选择“暂停”命令；如果需要恢复某一线程的执行，可以在“线程”窗口中右键单击该线程，然后选择“恢复”命令。

在“线程”窗口中右键单击某个线程，然后选择“转至源”命令，会在源代码编辑器中打开对应的代码。

3.7.4 “会话”窗口

“会话”窗口列出当前在 NetBeans 集成开发环境中运行的调试会话。选择“窗口”→“调试”→“会话”命令，可以打开“会话”窗口，如图 3-24 所示。

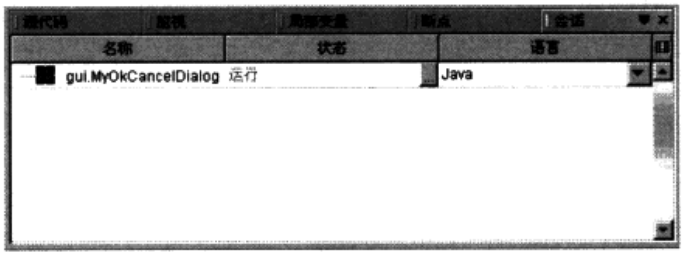


图 3-24 “会话”窗口

“会话”窗口的“状态”列中，给出了会话的状态，可有如下几种状态。

- 正在启动：程序正在启动。不能对应用程序进行暂停、恢复或单步执行。
- 运行：被调试的应用程序已经启动，并正在执行指令。
- 已停止：因为遇到断点、已将应用程序暂停或执行了步骤命令（步入、越过、步出或运行至光标处），所以应用程序暂停。

3.7.5 “源”窗口

“源”窗口中列出了调试器可能会使用的源代码文件。选择“窗口”→“调试”→“源”命令，可以打开“源”窗口，如图 3-25 所示。

Download at http://www.pin5i.com/

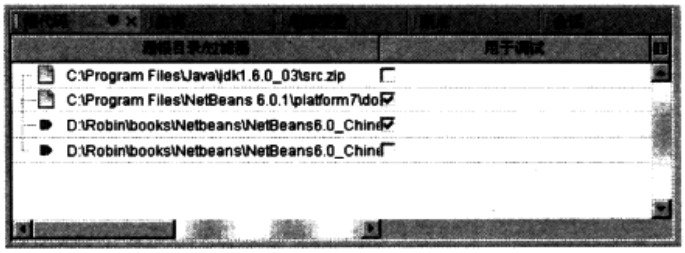


图 3-25 “源”窗口

“源”窗口中的“用于调试”列中列出了哪些源代码是用于调试的，哪些不是。可以通过单击该列来修改这些属性。

3.8 小结

调试技术是程序开发中必不可少的技术，可以帮助开发人员迅速找到问题。由于在后续章节的学习中，都有可能进行代码调试，因此提前进行介绍。

本章介绍了在 NetBeans 中的调试技术，主要包括调试会话中各个调试窗口的使用方法。通过本章的学习，应该掌握如下的内容：

- 断点的类型以及断点的设置和管理方法
- 如何进行单步跟踪程序
- 如何查看变量和表达式的值

Download at http://www.pin5i.com/

第4章

代码重构

在程序开发中经常需要修改代码，例如修改变量名。如果一个变量在不同文件的不同地方被使用，那么每个地方都需要修改。进行手工修改很容易发生错误，NetBeans 提供了代码重构的功能，可以帮助开发人员迅速地修改代码而不发生错误。

所谓代码重构，是指使用较小的转换来重新构造代码，而不会改变任何程序行为。NetBeans 集成开发环境的重构功能通过完成以下操作来简化代码重构的过程：确定要进行的更改，预览应用程序中受影响的部分（此时用户可以选择更改哪些内容），然后对代码进行必要的更改。例如，如果使用“重命名”操作来更改类名，集成开发环境将在代码中查找该名称的所有使用实例，并更改每个出现的名称。本章将主要介绍几种常用的代码重构方法，包括重命名、提取方法、更改参数、封装字段、移动类等。

本章主要内容

- 重命名类、接口、字段和方法
- 提取方法
- 更改参数
- 提取接口和超类

本章亮点

- 重命名类、接口、字段和方法
- 移动类以及类成员
- 封装字段

4.1 重命名

在开发中，如果需要对变量、方法等的名称进行修改，可以使用重命名操作。重命名操作包括重命名变量、重命名字段、重命名方法、重命名类或者接口、重命名包、重命名文件、重命名项目等。

4.1.1 重命名项目

要重命名一个项目，可以在“项目”窗口中右键单击项目节点，然后选择弹出菜单中的“重命名项目”命令，打开“重命名项目”对话框，如图 4-1 所示。

在“项目名称”文本框中输入项目的新名字。如果要同时更改项目所在文件夹的名字，则应该选中“同时重命名项目文件夹”复选框。单击“重命名”按钮，就会把选中的项目修改为指定的名字。

Download at http://www.pin5i.com/

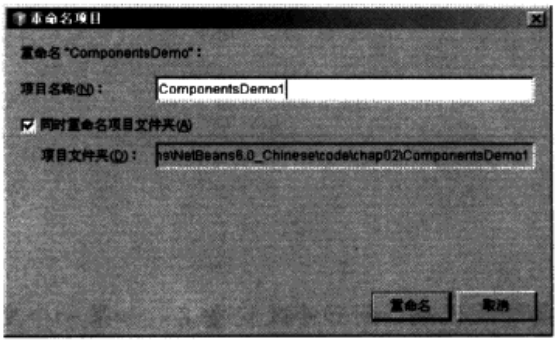


图 4-1 “重命名项目”对话框



4.1.2 重命名其他元素

重命名变量一般是在局部进行，因此也可以使用查找替换功能实现。一个类的某个字段或者变量有可能在不同的文件中被引用，因此如果使用查找替换功能，可能会产生错误，也比较麻烦。这时使用重命名操作会比较安全和快捷。

其他元素的重命名操作比较类似，因此这里只以重命名字段或者方法为例进行介绍。对字段或者方法进行重命名操作时，首先要在源代码编辑器中选中要更改的内容，如图 4-2 所示。然后通过下面几种方式打开重命名对话框，如图 4-3 所示：

- 选中“重构”→“重命名”命令；
- 在源代码编辑器中单击右键，选中弹出上下文菜单中的“重构”→“重命名”命令。

重命名对话框的标题可能会因为修改内容的不同而有所变化。这里把方法的名字由 name 修改为 bean_name。如果要同时修改注释中的名字，则要选中“对注释应用重命名”复选框。

```
/**
 *
 * @author Liu Bin
 */
public class MySimpleBean {
    private String name;

    /** Creates a new instance of MySimpleBean */
    public MySimpleBean() {
        name = "Hello World";
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

图 4-2 在源代码编辑器中选中要更改的内容

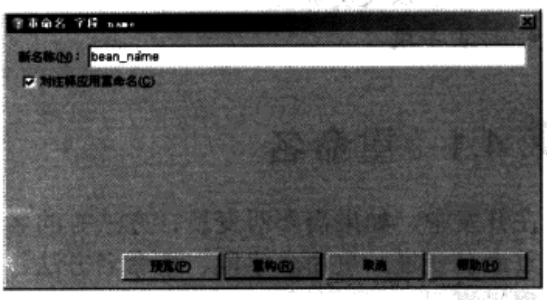


图 4-3 重命名对话框

对于一些复杂的项目，这里强烈建议开发人员先预览重命名的结果，因此单击“预览”按钮，打开“重构”窗口的物理视图，显示预览结果，如图 4-4 所示。

图 4-5 为 NetBeans 5.5 中“重构”窗口的物理视图，通过比较可以看到，在 NetBeans 6.0 中，“重构”窗口的功能有所增强，提供了在实际代码中的修改效果预览。

在“重构”窗口的物理视图中，给出了树形结构可能进行的重构操作。通过选择或清除相应的复选框，指定要应用重构更改的文件或代码元素。双击某行可以在源代码编辑器中打开它所引用的文件，并定位到相应的行上。

“重构”窗口左边的工具栏中有一些操作的按钮。“刷新重构数据”按钮 可以显示启动重构命令后已经执行的所有更改；“折叠/展开树中的所有节点”按钮 用于折叠或者展开树

Download at http://www.pin5i.com/

中的所有节点；“显示逻辑视图”按钮 用于显示如图 4-6 所示的逻辑视图；“显示物理视图”按钮 用于显示如图 4-5 所示的物理视图。

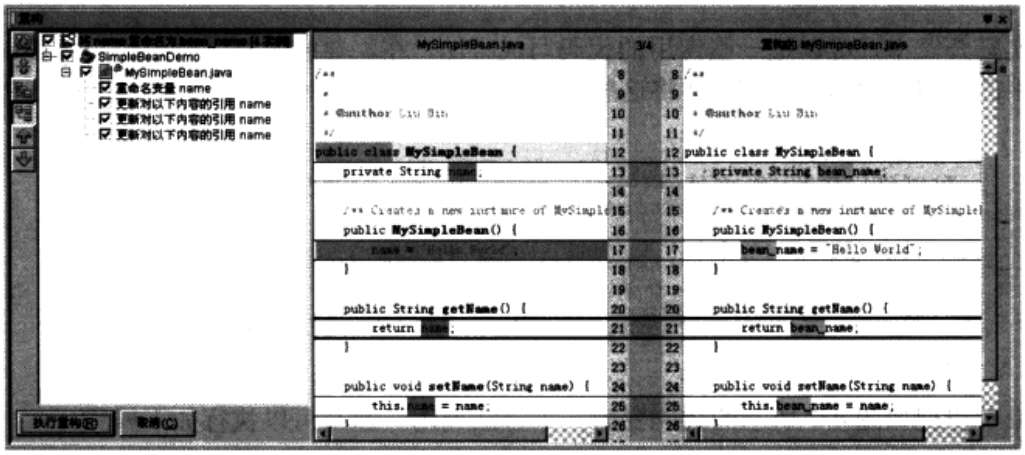


图 4-4 “重构”窗口的物理视图

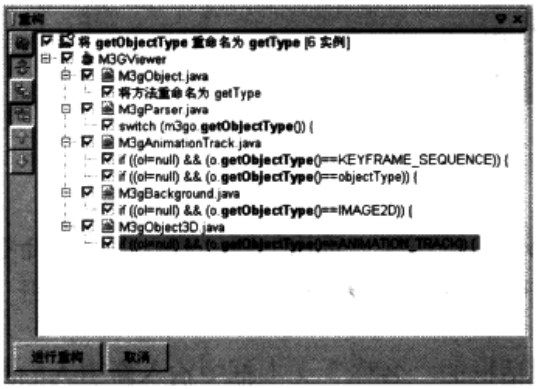


图 4-5 NetBeans 5.5 中“重构”窗口的物理视图

“上一个实例”按钮 用于定位到上一个重构实例，并在源代码编辑器中打开相应的代码行；“下一个实例”按钮 用于定位到下一个重构实例，并在源代码编辑器中打开相应的代码行。

在完成预览后，单击“重构”按钮后，将应用重构更改；单击“取消”按钮则取消重构操作。

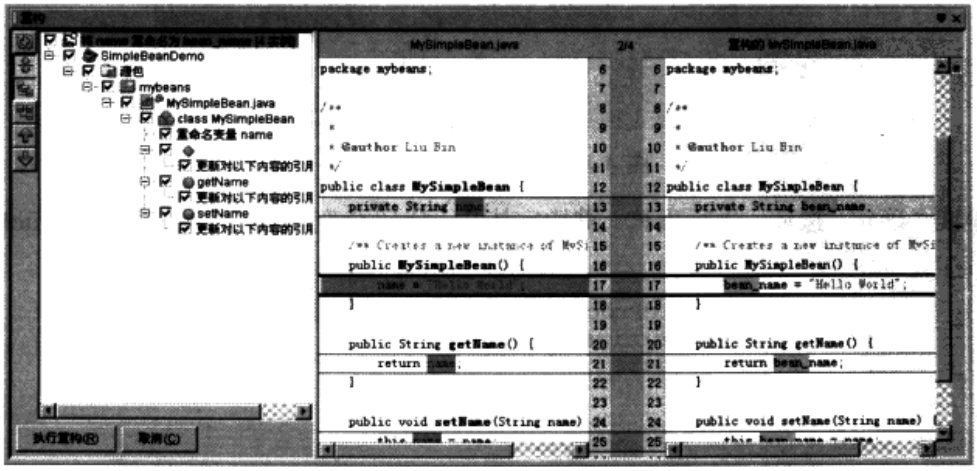


图 4-6 “重构”窗口的逻辑视图

Download at http://www.pin5i.com/

4.2 引入方法

在实际开发中，可以把一些常用的代码写成方法，在需要的时候进行调用。NetBeans 支持把方法中的一些代码进行提取，并将这些代码变为一个方法。例如，对于下面的 writeProperties 方法：

```
protected void writeProperties() throws M3gException {
    super.writeProperties();

    Object o = this.properites.get("Property ID");
    if (o != null && o instanceof Long) {
        this.propertyID = ((Long)o).longValue();
    }
}
```

如果希望把如图 4-7 所示的代码变为一个方法调用，那么可以执行如下操作：

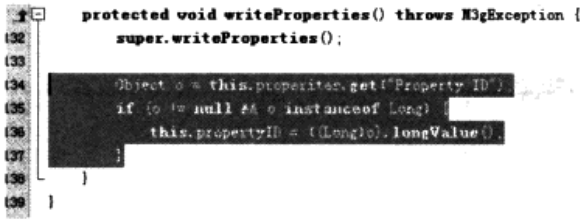


图 4-7 提取方法

❶ 选中要提取的代码，然后选择“重构”→“引入方法”命令，或者在源代码编辑器中单击右键，选择弹出菜单中的“重构”→“引入方法”命令，打开“引入方法”对话框，如图 4-8 所示。

❷ 在“引入方法”对话框的“名称”文本框中输入新引入的方法名。在“访问权限”中可以选择新方法的可访问性。

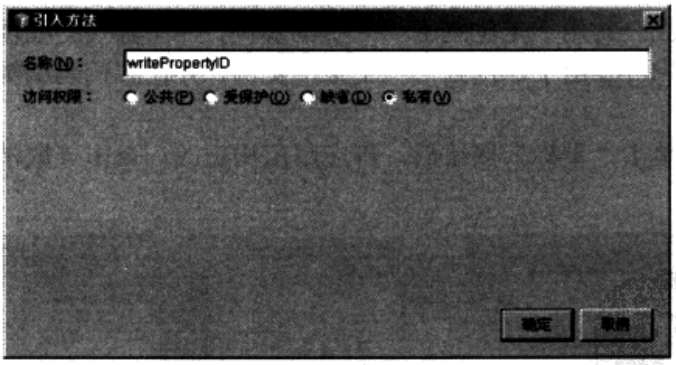


图 4-8 “提取方法”对话框

❸ 单击“确定”按钮，执行重构。重构完毕后，代码会变为如下所示，其中 writePropertyID 就是提取后的方法。

```
protected void writeProperties() throws M3gException {
    super.writeProperties();
    writePropertyID();
}

private void writePropertyID() {

    Object o = this.properites.get("Property ID");
```

Download at http://www.pin5i.com/


```
        if (o != null && o instanceof Long) {
            this.propertyID = ((Long)o).longValue();
        }
    }
```

4.3 更改方法的参数

在开发过程中，有时候需要修改某个方法的参数。如果进行手工修改，那么就必须把所有涉及这个方法的代码都进行修改。例如在类 A 中有一个 aaa 方法，类 B 继承于类 A，并覆盖了 aaa 方法。类 C 从类 B 派生，也覆盖了 aaa 方法。现在要把类 A 的 aaa 方法修改为 new_aaa，那么就需要同时修改类 B 和类 C 中的 aaa 方法。当然从语法上讲，如果不修改类 B 和类 C 中的 aaa 方法，也不会出现任何的语法错误。但是，逻辑上类 B 和类 C 中的 aaa 方法就不再和类 A 中的 new_aaa 有覆盖关系，从而导致意外的错误。这种错误一旦发生，很不容易找到原因。对于一个大的系统，如果手工修改方法的参数，将很容易发生问题。

NetBeans 的重构技术中提供了更改方法参数的功能，使得这个操作变得非常容易。例如要修改某个子类中的 write 方法，原代码如下所示。

```
protected void write(M3gOutputStream mos,
        java.util.Vector ref_table) throws java.io.IOException {
    uper.write(mos, ref_table);

    ...
}
```

write 方法覆盖了父类的 write 方法。并且在整个类家族中，都实现了这个 write 方法。现在要为这个方法添加一个参数，则具体的操作步骤介绍如下。

❶ 在源代码编辑器中选中 write 方法，然后选择“重构”→“更改方法参数”命令，或者在源代码编辑器中单击右键，选择弹出菜单中的“重构”→“更改方法参数”命令，打开“更改方法参数”对话框，如图 4-9 所示。在参数列表中，选中某个参数，单击“删除”按钮即可删除该参数。单击“上移”按钮可以把参数的位置提前一个位置；单击“下移”按钮可以把参数的位置置后一个位置。

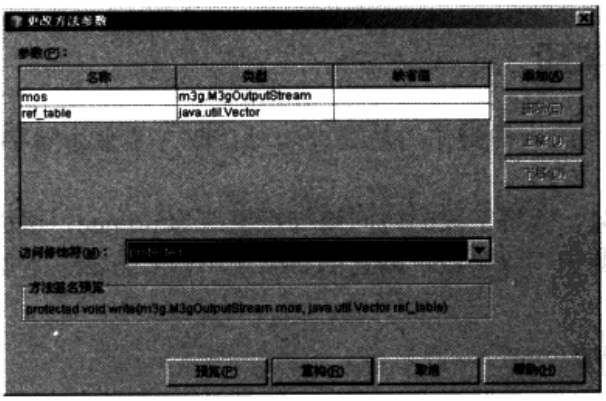


图 4-9 “更改方法参数”对话框

❷ 这里单击“添加”按钮，然后在“参数”列表中添加如图 4-10 所示的参数。在“名称”列中输入参数的名字；在“类型”列中输入参数的数据类型；在“值”列中添加参数的默认值。

❸ 在“可视性修饰符”下拉列表框中修改方法的可访问性。如果单击“重构”按钮，集成开发环境将跳过预览并自动应用更改。这里单击“预览”按钮，预览所有更改，打开“重构”窗口，如图 4-11 所示。

Download at http://www.pin5i.com/

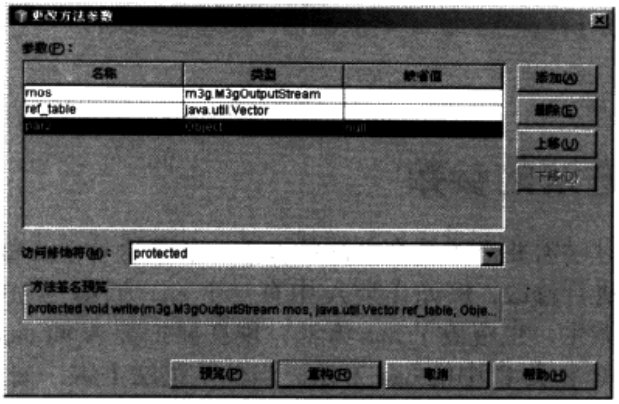


图 4-10　为方法添加一个参数

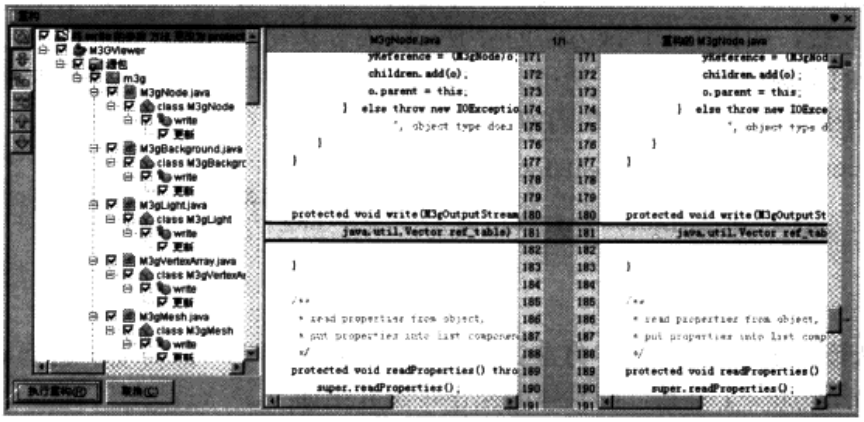


图 4-11　预览更改方法参数

❹ 从图 4-11 中可以看出，集成开发环境把所有要进行修改的 write 方法都列了出来。请检查列表，并清除任何不想更改的代码前的复选框（重构时将不操作该行的代码）。单击“执行重构”按钮完成方法参数的添加。

4.4　封装字段

字段封装是重构代码的一种操作，是通过一对存取方法来访问字段。存取方法也称为读/写方法或 getter 和 setter 方法。

通常，在封装字段时，会将字段的访问修饰符更改为 private，这样就无法从类外部直接引用该字段。如果其他类要引用该字段，则它们必须使用存取方法。下面的 MyClass 类中，包含一个共有的 name 属性。

```
public class MyClass {
    public String name = "";
    int age;

    public MyClass() {
    }
}
```

在源代码编辑器（或者“项目”窗口）中右键单击 name 字段。然后选择“重构”→“封装字段”命令，或者在源代码编辑器中单击右键，选择弹出菜单中的“重构”→“封装字段”命令，打开“封装字段”对话框，如图 4-12 所示。

Download at http://www.pin5i.com/

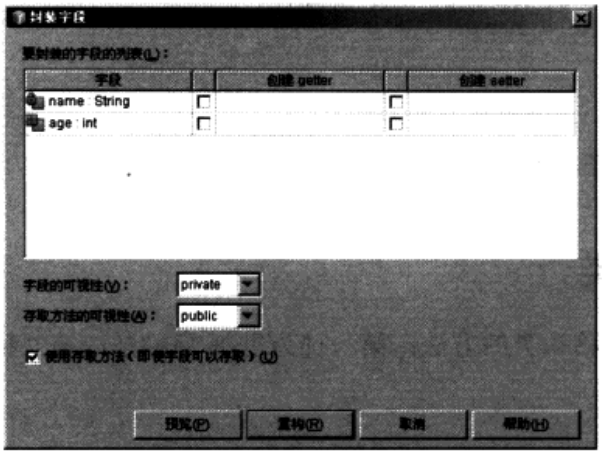


图 4-12 “封装字段”对话框

在“要封装的字段的列表”中，包含了类中字段的列表，选定要封装的字段的复选框。这里，要修改 name 字段，因此把 name 字段的 getter 和 setter 复选框都选中。

“字段的可视性”下拉列表框用于选择字段的可访问性修饰符，一般情况下都选择 private 修饰符。

“存取方法的可视性”下拉列表框用于选择存取方法（getter 和 setter）的可访问性修饰符。

如果选中“使用存取方法（即使字段可以存取）”复选框，那么将更新代码中对字段的所有直接引用，以使用存取方法。如果未选定此复选框，则不会替换代码中已包含的对字段的任何直接引用。

单击“预览”按钮，打开“重构”窗口，如图 4-13 所示。

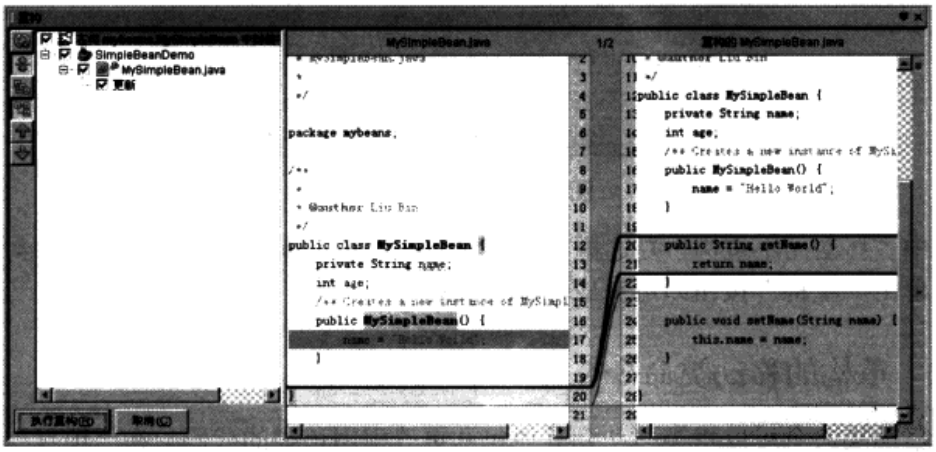


图 4-13 预览封装字段

单击“执行重构”按钮完成字段的封装。封装 name 字段后的 MyClass 类代码如下所示。

```
public class MyClass {
    private String name = "";
    int age;

    public MyClass() {
    }

    public String getName() {
        return name;
    }
```

Download at http://www.pin5i.com/


```
        public void setName(String name) {
            this.name = name;
        }
    }
```



注意：封装字段只能对任何非静态字段进行操作。



4.5 移动类

NetBeans 支持两种移动类的方法：第一种为复制/粘贴法；第二种是使用重构的移动类命令。下面分别进行介绍。



4.5.1 使用复制/粘贴法移动类

在“项目”窗口中选中要移动的类，然后右键单击该类，选择弹出菜单中的“剪切”命令（如果要复制该类，则应该选择“复制”命令）。

在要被复制到的项目包上单击右键（注意，一定是某个目标包上），选择弹出菜单中的“移动”命令，则直接会把类移动到目标包中。如果选择弹出菜单中的“重构移动”命令，会打开“移动类”对话框，如图 4-14 所示。

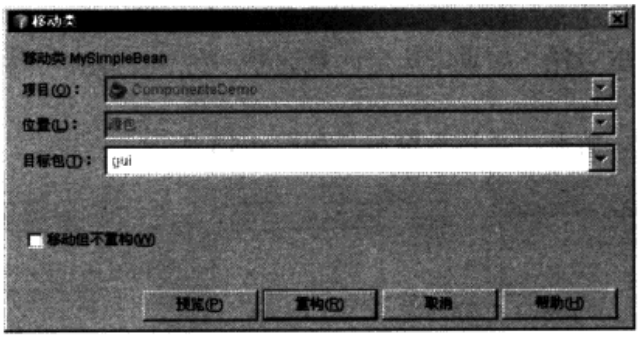


图 4-14 复制/粘贴类

如果选中“移动而不重构”复选框，那么将不执行重构操作，直接把类移动到目标项目的目标包中。单击“重构”按钮即可完成移动类的操作。



4.5.2 重构的移动类命令

重构的移动类命令可以将类移动到其他包并更改引用该类的代码。在“项目”窗口或“源代码编辑器”窗口中选中要移动的类，然后选择“重构”→“移动”命令，或者在源代码编辑器”窗口中单击右键，选择弹出菜单中的“重构”→“移动”命令，打开“移动类”对话框，如图 4-15 所示。

在“项目”下拉列表框中，可以选择要将类移动到的项目名称。在“位置”下拉列表框中可以选择要将类移动到目标项目的哪个部分，此处的正确值通常为“源包”，但也可以在此字段中表示其他源根目录。“目标包”下拉列表框用于选择要将类移动到的包。当要移动多个类时（执行移动类命令前选中了多个类），则会显示如图 4-16 所示的对话框。

在“类的列表”中显示了要移动的类的名称。如果只移动一个类，则不会显示此列表。单击“预览”按钮显示如图 4-17 所示的“重构”窗口。

Download at http://www.pin5i.com/

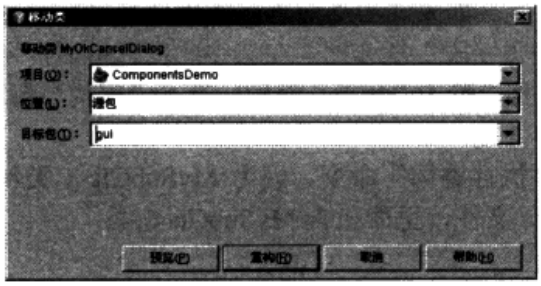


图 4-15 “移动类”对话框

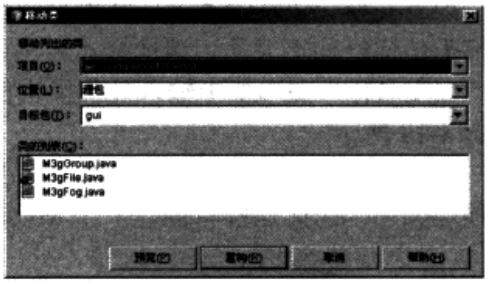


图 4-16 移动多个类时的“移动类”对话框

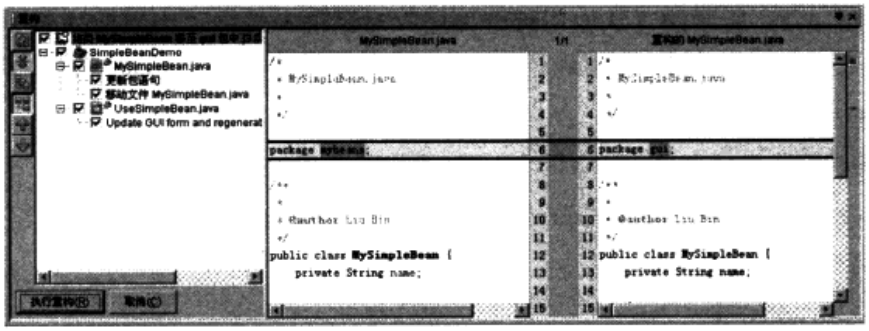


图 4-17 预览类的移动

在“重构”窗口中，确认重构的操作没有问题后，单击“执行重构”按钮，即可完成类的移动操作。

4.6 从内层移至外层

重构的“从内层移至外层”命令，可以将内部类在分层结构中上移一级。如果选定的类直接嵌套在顶层类中，则会创建一个新的顶层类。如果选定的类嵌套在内部类中，则将选定类移动到嵌套了该类的内部类的层中。例如，下面的代码中 MySubClass 类为 MyClass 的内部类。

```
public class MyClass {
    public String name = "";

    class MySubClass {

    }
}
```

在源代码编辑器中选中 MySubClass 类，然后选择上下文菜单中的“重构”→“从内层移至外层”命令，打开“从内层移至外层”对话框，如图 4-18 所示。

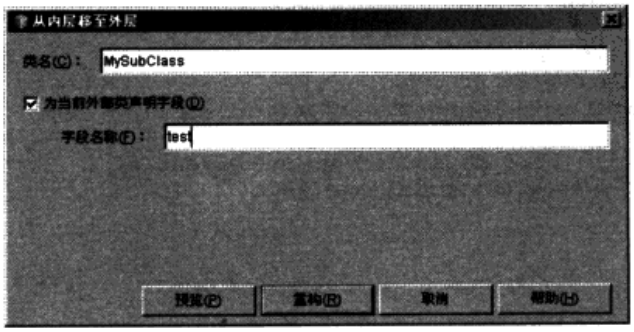


图 4-18 “从内层移至外层”对话框

Download at http://www.pin5i.com/

如有必要可以在“类名”文本框中更改类的名字。如果要为当前外部类生成实例字段并将外部类传递给构造方法，请选择“为当前外部类声明字段”复选框。在选中此复选框后，还要在“字段名称”文本框中输入外部类实例字段的名称。单击“预览”按钮，打开“重构”窗口，如图 4-19 所示。

在“重构”窗口中，确认操作无误后，单击“执行重构”命令。因为 MySubClass 类包含在顶层类中，因此这样将创建一个 MySubClass.java 文件，里面包含 MySubClass 类：

```
class MySubClass {

}
```

如果在如图 4-18 所示的“从内层移至外层”对话框中选中“为当前外部类声明字段”复选框，并在“字段名称”文本框中输入“test”，那么创建的 MySubClass 类代码如下所示：

```
class MySubClass {

    MyClass test;

    MySubClass(MyClass test) {
        super();
        this.test = test;
    }
}
```

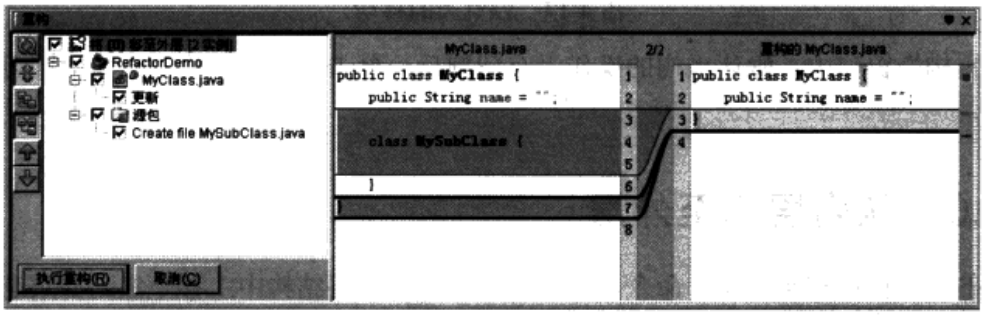


图 4-19　预览把类从内层移至外层

4.7　将匿名类转换为外部类

重构的将匿名类转换为外部类命令，可以把一个匿名类提取出来，并保存为一个外部类。在使用此操作时，将创建一个新的内部类，并将匿名内部类替换为对新内部类的调用。例如在如下的方法中，包含了一个 MouseAdapter 的匿名类。

```
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JPanel;

public class MyClass extends JPanel {
    void addListener() {
        addMouseListener(new MouseAdapter() {

            public void mouseClicked(MouseEvent e) {
                System.out.println(e.getX());
            }
        });
    }
}
```

在源代码编辑器中，右键单击 MouseAdapter 类，然后选择弹出菜单中的“重构”→“将

Download at http://www.pin5i.com/

匿名类转换为外部类”命令，NetBeans 集成开发环境会自动把 MouseAdapter 改变为一个内部类，代码如下所示：

```
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JPanel;

public class MyClass extends JPanel {
    void addListener() {
        addMouseListener(new MouseAdapterImpl());
    }

    private static class MouseAdapterImpl extends MouseAdapter {

        public MouseAdapterImpl() {
        }

        public void mouseClicked(MouseEvent e) {
            System.out.println(e.getX());
        }
    }
}
```

4.8 提取接口

重构的提取接口命令可以通过某个类或接口中的选定非静态公共方法来创建新接口。例如，下面的类中包含一个构造函数和三个公有的方法：getName、setName 和 getTransparency。其中，getTransparency 是 java.awt.Transparency 接口中定义的方法。

```
public class MyClass implements java.awt.Transparency {
    private String name = "";

    public MyClass() {
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getTransparency() {
        return 0;
    }
}
```

在源代码编辑器的文件中单击鼠标右键，然后选择“重构”→“提取接口”命令，打开“提取接口”对话框，如图 4-20 所示。

在“接口名称”文本框中输入接口的名称，这里输入 MyNewInterface。在“要提取的成员”列表中，选择要提取到新接口中的成员。

如果从中提取接口的类已实现某个接口，这里实现了 Transparency 接口。在“要提取的成员”列表中，还会包含实现该接口的项。如果选中 implements java.awt.Transparency 复选框，则新提取的接口会继承这个接口。

单击“预览”按钮，打开“重构”窗口，如图 4-21 所示。

Download at http://www.pin5i.com/

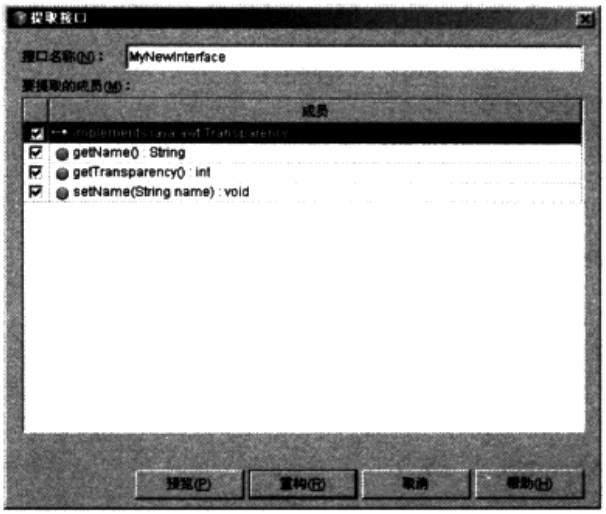


图 4-20 “提取接口”对话框



图 4-21 预览提取接口

在“重构”窗口中，确认操作无误后，单击“执行重构”命令完成接口的提取。集成开发环境将创建一个新的文件 MyNewInterface.java，里面包含 MyNewInterface 接口：

```
import java.awt.Transparency;

public interface MyNewInterface extends Transparency {
    String getName();

    int getTransparency();

    void setName(String name);

}
```

MyClass 类则变为如下形式的代码，其实现接口由 Transparency 变为 MyNewInterface。

```
public class MyClass implements MyNewInterface {
    private String name = "";

    public MyClass() {
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getTransparency() {
        return 0;
    }
}
```

在如图 4-20 所示的“提取接口”对话框中，如果不选中 implements java.awt.Transparency 复选框，那么最后生成的 MyNewInterface 接口代码如下所示：

```
public interface NewInterface {
    String getName();

    int getTransparency();

    void setName(String name);
```

Download at http://www.pin5i.com/


```
}
```

MyClass 类的代码如下所示：

```
public class MyClass implements java.awt.Transparency, NewInterface {
    private String name = "";

    public MyClass() {
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getTransparency() {
        return 0;
    }
}
```



4.9 小结

本章主要介绍如何在 NetBeans 集成开发环境中使用重构命令完成代码的快速修改。本章介绍的内容不是编程必需的，但是学习掌握后可以大大提高编程的效率。对于有一定编程基础的读者，这部分应该作为提高部分好好掌握。通过本章的学习，读者应该掌握以下内容：

- 如何重命名项目、字段、方法等
- 如何利用提取方法命令，把一部分代码提取为方法
- 如何快速安全地修改方法的参数
- 如何对字段进行封装，生成字段的读/写方法
- 如何把一个类移动到另外的项目或者包中
- 如何把类由内部移动到上一层
- 如何把匿名类提取为一个内部类
- 如何从一个类中提取接口

Download at http://www.pin5i.com/

第5章

JavaBean 开发

代码复用一直是开发人员追求的一个目标，组件技术是实现代码复用的一个很好途径。目前流行的组件技术有 COM、DCOM、CORBA 等。JavaBean 是利用 Java 实现的组件技术，具有 Java 语言所有的特点，如易用性以及平台无关性。JavaBean 的设计目标就是一次性编写、任何地方执行、任何地方重用。

本章主要介绍如何在 NetBeans 中开发和利用 JavaBean 组件。首先介绍 JavaBean 的基本知识，然后通过实例介绍如何开发一个简单的 JavaBean，以及如何使用这个 JavaBean 组件。接着，逐步深入给出可视化 JavaBean 的开发方法，如何为 JavaBean 添加属性等内容。在学习本章前，建议读者先认真学习一下 JavaBean 开发的相关知识。

本章主要内容

- JavaBean 的基本概念
- 如何创建和使用一个简单的 JavaBean
- JavaBean 的属性
- 如何使用下拉列表框形式的属性编辑器
- 如何绘制属性
- 如何自定义属性编辑器

本章亮点

- 可视化 JavaBean 的创建和使用方法
- 如何在“属性”编辑器中绘制属性
- 如何定制自己的属性编辑器

5.1 JavaBean 概述

在软件开发中，组件技术是非常重要的一项技术。利用组件，开发人员可以像搭积木一样把软件功能作为一个组件来快速地组装成一个应用程序。组件技术的一个优点就是可以复用，即一个组件可以被不同的系统和平台使用。JavaBean 是一种专门为当前开发人员设计的全新的组件技术，它为软件开发者提供了一种极佳的问题解决方案。JavaBean 具有如下的特点。

- 完全的可移植性：JavaBean 是使用 Java 语言实现的，众所周知，Java 语言具有很好的可移植性和跨平台性，因此这为 JavaBean 提供了独立于平台的组件解决方案。JavaBean 可以在任何支持 Java 的系统中无须修改地执行，提供了高度的可移植性。
- 分布式计算支持：JavaBean 提供了分布式计算的能力，使得开发者可以在任何时候使用分布式计算机制。
- 应用程序构造器支持：JavaBean 体系结构支持指定设计环境属性和编辑机制，这使得开发者可以使用可视化应用程序构造器无缝地组装和修改 JavaBean 组件。例如在 NetBeans 集成开发环境下，就可以可视化地创建一个 JavaBean 组件。

Download at http://www.pin5i.com/

JavaBean 组件技术不仅可以应用于传统的桌面应用程序开发，还可以与 JSP 配合开发 Web 应用程序，以及在企业级开发中（Enterprise JavaBean）使用。

NetBeans 中提供了对开发 JavaBean 组件的很好支持，可以快速地创建并使用 JavaBean 组件。在下面各节中将详细介绍在 NetBeans 中开发 JavaBean 的方法。

5.2 如何创建一个简单的 JavaBean

其实，JavaBean 也是普通的 Java 对象，只不过它遵循了一些特别的约定而已。NetBeans 集成开发环境提供了创建 JavaBeans 组件的模板。这些模板为非可视 Bean 提供了框架代码。也可以使用 JFC/Swing 模板来创建可视 Bean。

5.2.1 创建一个简单的 JavaBean

在 NetBeans 中，可以把 JavaBean 添加到组件面板中，以便于代码重复利用。下面是创建一个简单 JavaBean 的过程。

❶ 创建一个 Java 标准项目，项目的名字为“SimpleBeanDemo”。

❷ 创建一个包，名字为“mybeans”。

❸ 选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”窗口中选择“Java”节点，如图 5-1 所示。

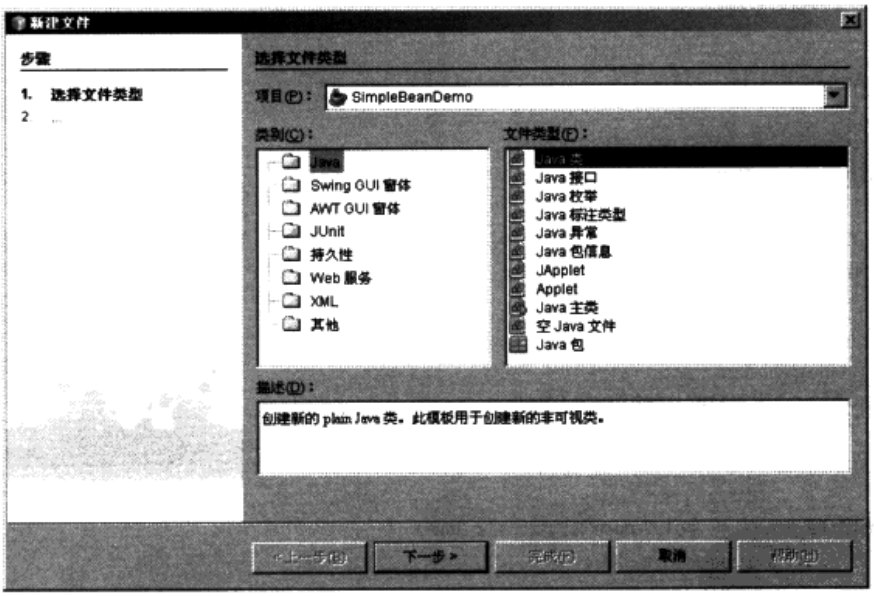


图 5-1 “新建文件”对话框的“Java 类”类别

❹ 在“文件类型”窗口中选择“Java 类”，然后单击“下一步”按钮，打开“新建 Java 类”对话框，如图 5-2 所示。

❺ 在“类名”文本框中输入“MySimpleBean”，在“包”下拉列表框中选择包“mybeans”。单击“完成”按钮，创建一个 Java 类，也就是创建一个 JavaBean。

❻ 在源代码编辑器中，为 MySimpleBean 类添加一个 String 类型的 name 属性，并添加一个 getName 方法用于读取 name 的值；添加一个 setName 方法设置 name 的值。具体实现代码如下所示：

Download at http://www.pin5i.com/

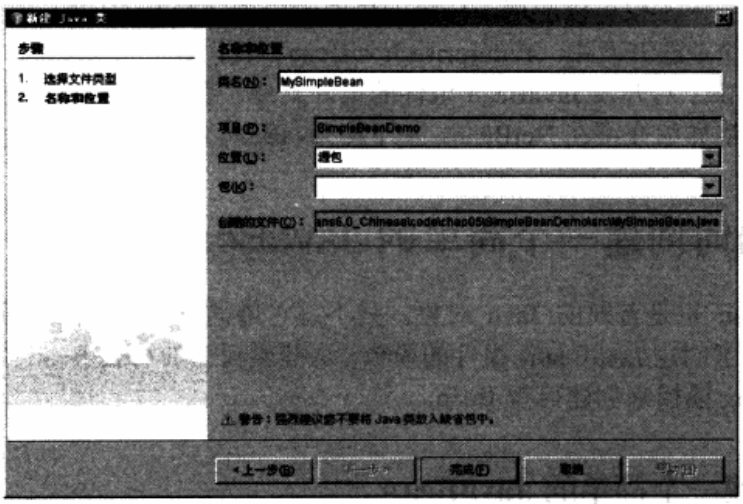


图 5-2 “新建 Java 类”对话框

```
/*
 * MySimpleBean.java
 *
 */

package mybeans;

/**
 *
 * @author Liu Bin
 */
public class MySimpleBean {
    private String name;

    /** Creates a new instance of MySimpleBean */
    public MySimpleBean() {
        name = "Hello World";
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

❼ 在“项目”窗口中选中“MySimpleBean.java”文件，然后选择“工具”→“添加到组件面板”命令，打开“选择组件面板类别”对话框，如图 5-3 所示。

❽ 选中“Bean”类别，单击“确定”按钮，把这个 JavaBean 添加到“Bean”类别中。此时，会在“组件面板”窗口中看到刚刚创建的 MySimpleBean，如图 5-4 所示。至此，完成了一个简单的 JavaBean 的创建。



说明：新创建的 Bean 类需要经过编译以后才能使用。

Download at http://www.pin5i.com/

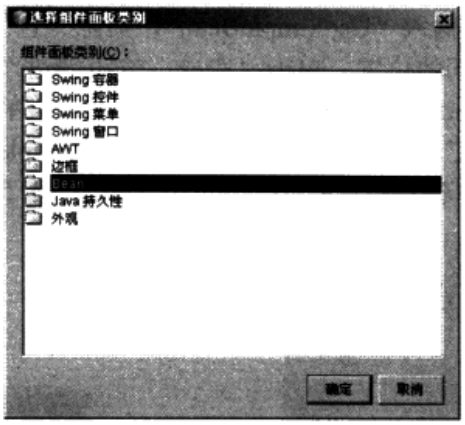


图 5-3 “选择组件面板类别”对话框



图 5-4 把 MySimpleBean 组件添加到了组件面板

5.2.2 如何使用 JavaBean

上面介绍了如何创建一个简单的 JavaBean，下面具体演示如何在 Java 项目中使用这个 JavaBean。

❶ 确认上一节中创建的项目“SimpleBeanDemo”为主项目，如果不是主项目就设置为主项目。

❷ 选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”窗口中选择“Swing GUI 窗体”节点，在“文件类型”窗口中选择“JFrame”窗口。单击“下一步”按钮，打开“新建 JFrame 窗体”对话框，如图 5-5 所示。

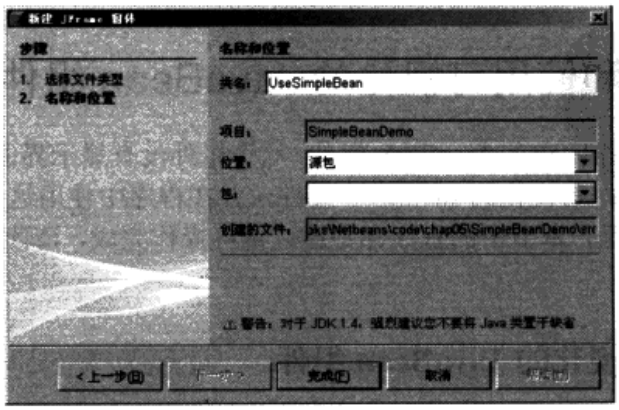


图 5-5 “新建 JFrame 窗体”对话框

Download at http://www.pin5i.com/

❸ 在“类名”文本框中输入“UseSimpleBean”，清除“包”下拉列表框中的内容，让这个窗体创建在与 JavaBean 不同的位置。单击“完成”按钮创建“JFrame”窗体。

❹ 在 GUI 设计器中，右键单击 JFrame 窗体，选择弹出菜单中选择“从组件面板上添加”→“Bean”→“MySimpleBean”命令，如图 5-6 所示。把 MySimpleBean 添加到这个窗体 JFrame 后，可以在“检查器”窗口中看到添加的 MySimpleBean，如图 5-7 所示。

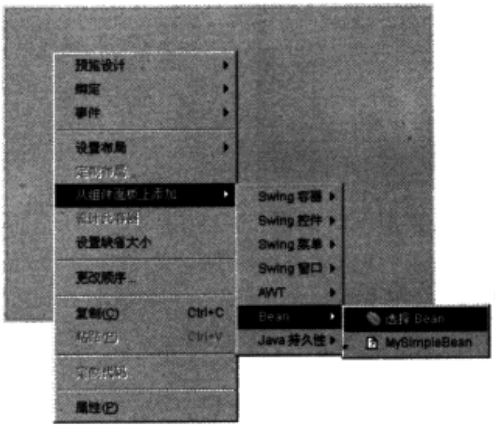


图 5-6 添加 MySimpleBean 组件

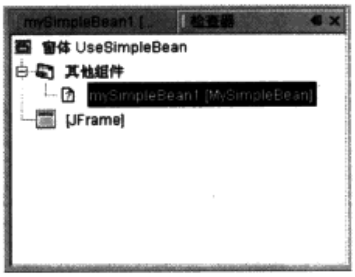


图 5-7 “检查器”窗口中的 MySimpleBean

❺ 选中这个 JavaBean 组件后，可以在“属性”窗口中修改组件的 name 属性，如图 5-8 所示。

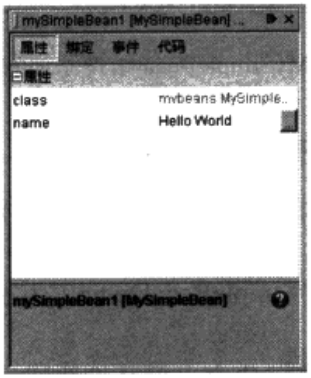


图 5-8 “属性”窗口修改组件的属性

至此，就可以在这个新的程序中使用创建的 JavaBean 组件了。

5.3 创建和使用可视化的 JavaBean 组件

上一节中创建了一个简单的 JavaBean 组件，这个组件没有显示界面，只有一个 name 属性。本节将介绍如何创建一个可视化的 JavaBean 组件，并在程序中使用这个可视化的 JavaBean 组件。同时，值得一提的是，本节中的程序都是通过鼠标操作完成，请读者注意 GUI 设计器中连接模式的使用方法。

5.3.1 创建可视化的 JavaBean 组件

在 Swing 组件中，有一个进度栏（JProgressBar）组件。该组件可以在进度条的中央显示当前的进度值，如图 5-9 所示。

Download at http://www.pin5i.com/



图 5-9 JProgressBar 组件

JProgressBar 的进度显示值，只能显示在进度条的中央，不能在其他位置。下面将创建一个可视化的 JavaBean，把进度值显示在其他的位置，其操作步骤介绍如下。

❶ 创建一个 Java 标准项目，项目的名字为“GuageBean”。

❷ 创建一个包，名字为“guagebean”。

❸ 选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”窗口中选择“Swing GUI 窗体”节点，如图 5-10 所示。

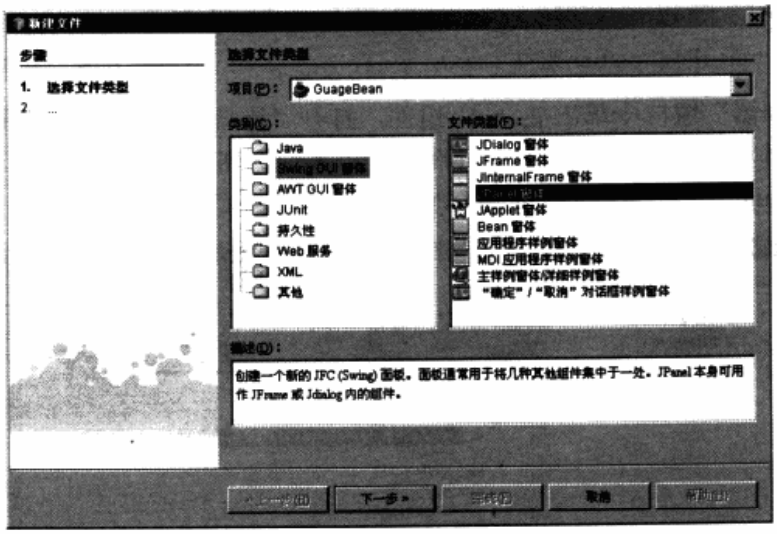


图 5-10 “新建文件”对话框

❹ 在“文件类型”窗口中选择“JPanel 窗体”，单击“下一步”按钮，打开“新建 JPanel 窗体”对话框，如图 5-11 所示。

❺ 在“类名”文本框中输入“MyGuage”，在“包”下拉列表框中选择“guagebean”包。单击“完成”按钮，创建一个可视化的 JavaBean。

❻ 在 GUI 设计器中添加一个 JLabel 组件，这个组件用于显示进度条的值。双击 JLabel 组件，把 text 值修改为 0，如图 5-12 所示。

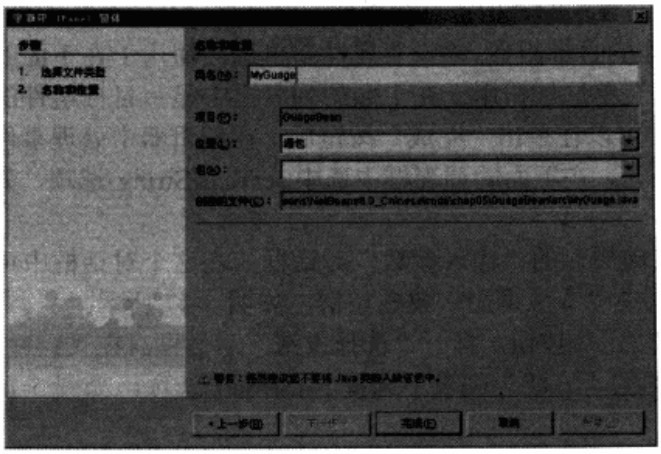


图 5-11 “新建 Jpanel 窗体”对话框



图 5-12 添加一个 JLabel 组件

Download at http://www.pin5i.com/

❼ 在 GUI 设计器中添加一个 JProgressBar 组件，在“属性”窗口中把 stringPainted 属性选中。此时会在进度条的中央显示进度值字符串，如图 5-13 所示。

❽ 在 GUI 设计器中，选中工具栏中的“连接模式”按钮。单击“检查器”窗口中的 jProgressBar1 组件，选择事件源组件。此时在 GUI 设计器中的进度条组件会变为红色，如图 5-14 所示。

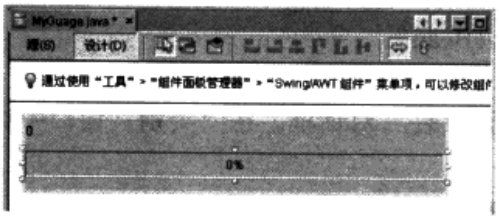


图 5-13 添加一个 JProgressBar 组件

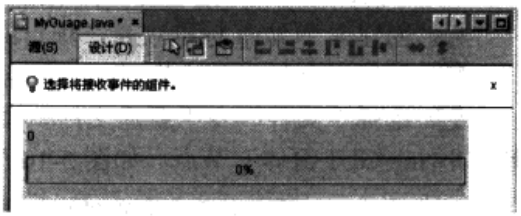


图 5-14 选择事件源组件

❾ 在“检查器”窗口中单击 jLabel1 组件，打开“连接向导”的“选择源事件”对话框。在“源组件”文本框中给出了事件的源组件，在“事件”列表中可以选择源事件。在“事件”列表中选择“change”→“stateChanged [jProgressBar1StateChanged]”节点，如图 5-15 所示。

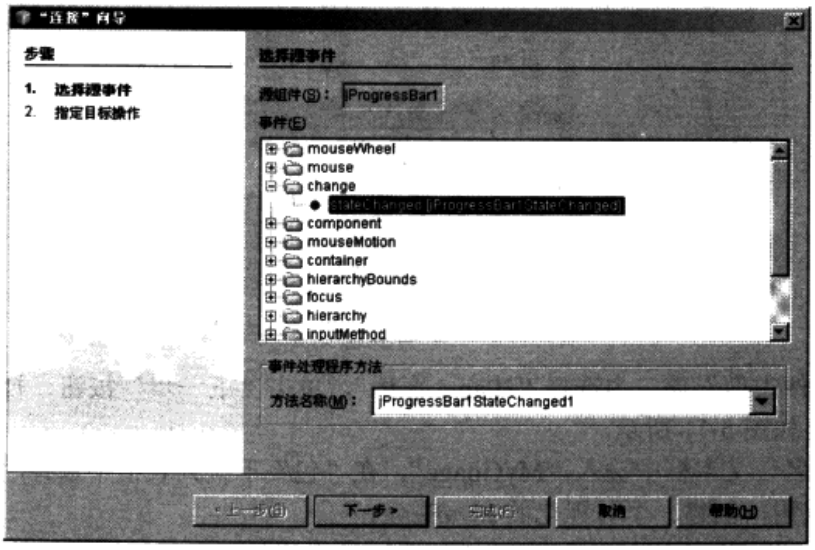


图 5-15 “连接向导”的“选择源事件”对话框

❿ 单击“下一步”按钮，打开“连接向导”的“指定目标操作”对话框。在“目标组件”文本框中显示了源组件发生源事件时要操作的目标组件。“设置属性”单选按钮用于在下面的列表框中显示目标组件的属性；“方法调用”单选按钮用于在下面的列表框中显示目标组件的成员方法；选中“用户代码”单选按钮后，可以在单击“完成”按钮后，在编辑器中处理事件的定制代码。这里选中“方法调用”单选按钮，在下面的列表框中选中 setText(String)选项，如图 5-16 所示。

⓫ 单击“下一步”按钮，打开“连接向导”的“输入参数”对话框。在这个对话框中可以选择目标组件的方法调用的参数。这里选中“方法调用”单选按钮，如图 5-17 所示。

⓬ 单击“方法调用”单选按钮旁边的“...”按钮，打开“选择方法”对话框。在“组件”下拉列表框中选中“jProgressBar1”，在下面的“方法”列表中选择 getString()选项，如图 5-18 所示。

Download at http://www.pin5i.com/

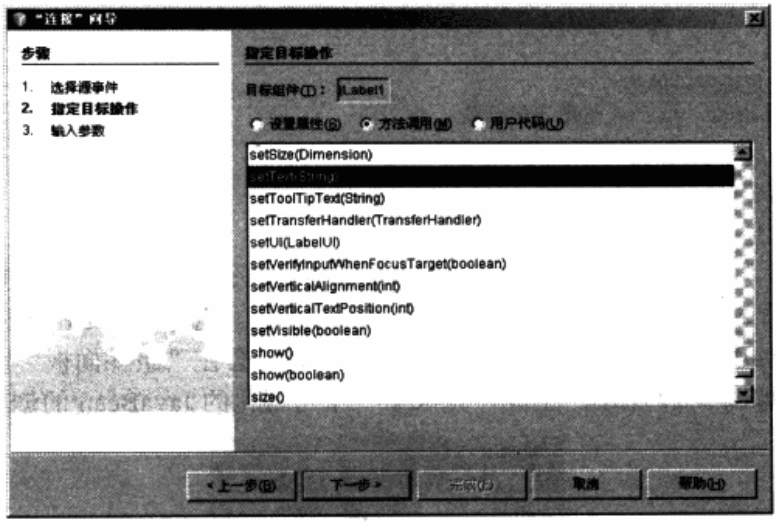


图 5-16 “连接向导”的“指定目标操作”对话框

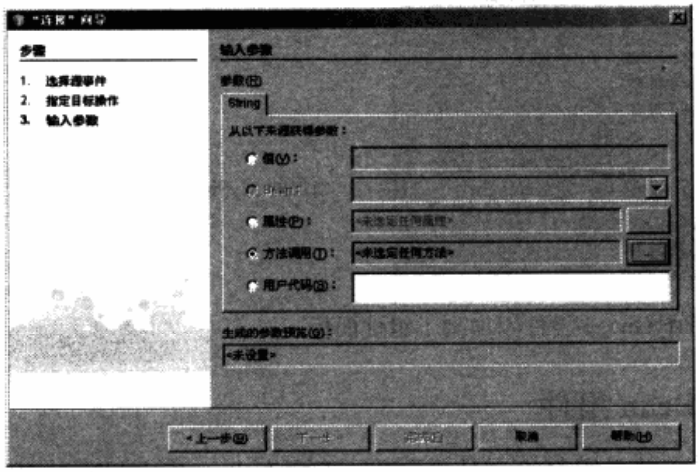


图 5-17 “连接向导”的“输入参数”对话框

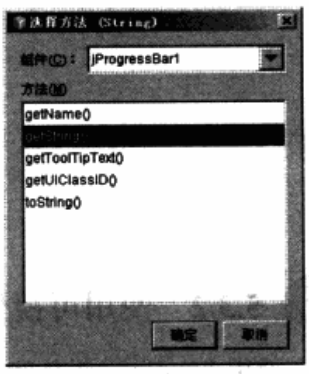


图 5-18 “选择方法”对话框

⓭ 单击“确定”按钮，返回“输入参数”对话框，单击“完成”按钮，完成代码的自动添加。这里添加了如下所示的代码：

```
jProgressBar1.addChangeListener(new javax.swing.event.ChangeListener() {
        public void stateChanged(javax.swing.event.ChangeEvent evt) {
            jProgressBar1StateChanged(evt);
        }
});

…
private void jProgressBar1StateChanged(javax.swing.event.ChangeEvent evt) {
    jLabel1.setText(jProgressBar1.getString());
 }
```

⓮ 在源代码编辑器中，添加如下所示的代码。其中，getValue()方法用于返回进度条的进度值；setValue()方法用于设置进度条的进度值。

```
/**
 * 返回进度条的值
 * @return 返回进度条的当前进度值
 */
public int getValue() {
    return jProgressBar1.getValue();
```

Download at http://www.pin5i.com/


```
}

/**
 * 设置进度条的值
 * @param value 进度条的新值
 */
public void setValue(int value) {
    jProgressBar1.setValue(value);
}
```

⑮ 生成项目。在“项目”窗口中选中“MyGuage.java”文件，然后选择“工具”→“添加到组件面板”命令，打开“选择组件面板类别”对话框。选中“Bean”类别，单击“确定”按钮，把这个 JavaBean 添加到“Bean”类别中。此时，会在“组件面板”窗口中看到刚刚创建的 MyGuage，如图 5-19 所示。至此，完成了一个可视化的 JavaBean 的创建。

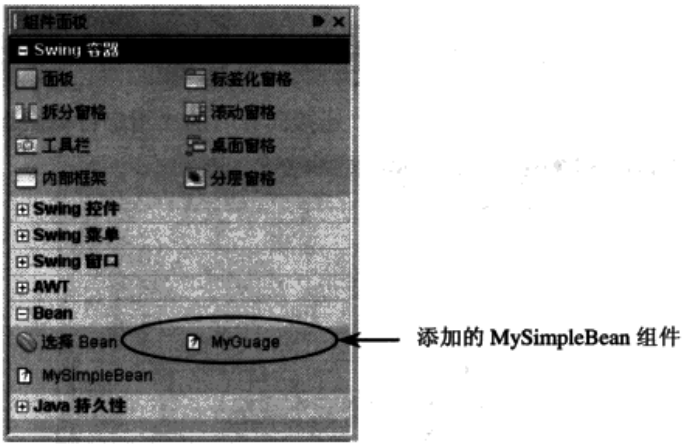


图 5-19 把 MyGuage 组件添加到了组件面板

5.3.2 使用可视化的 JavaBean 组件

使用 JavaBean 组件和使用一般的 Swing 组件没有什么区别。下面在上一节的“GuageBean”项目中，仿照第 5.2.2 节的方法创建一个 JFrame，名字为“MyGuageTest”。在 JFrame 中添加 MyGuage 组件。再添加一个 JButton，修改 text 属性为“更新”，如图 5-20 所示。

在 GUI 设计器中，选中工具栏中的“连接模式”按钮。单击“导航”窗口中的 jButton1 组件，选择事件源组件。再单击 myGuage1 组件，打开“连接向导”的“选择源事件”对话框。在“事件”列表中选择“action”→“actionPerfomed [jButton1ActionPerformed]”节点，如图 5-21 所示。

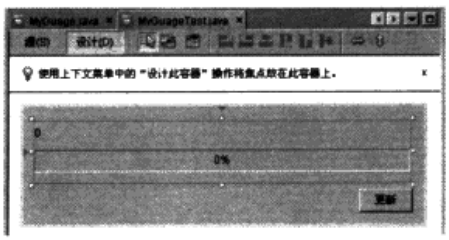


图 5-20 在 JFrame 中添加 MyGuage 组件

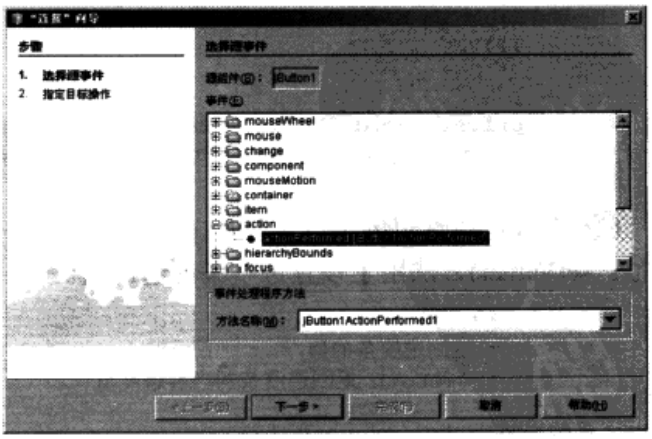


图 5-21 “选择源事件”对话框

Download at http://www.pin5i.com/

单击“下一步”按钮，打开“连接向导”的“指定目标操作”对话框。选中“方法调用”单选按钮，在下面的列表框中选中 setValue(int)选项，如图 5-22 所示。这里的 setValue 方法就是上一节中添加的方法。

单击“下一步”按钮，打开“连接向导”的“输入参数”对话框。选中“值”单选按钮，并在后面的文本框中输入“myGuage1.getValue()+1”，如图 5-23 所示。

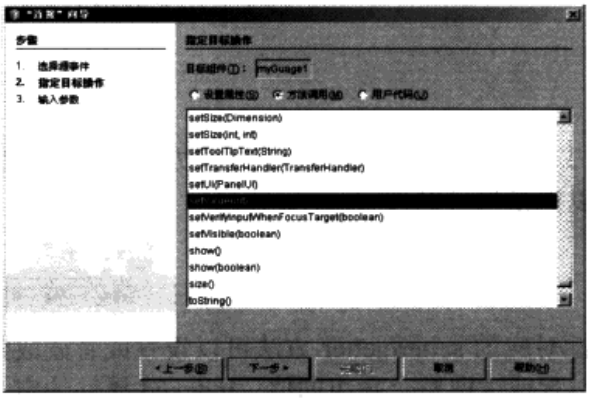


图 5-22 “选择目标操作”对话框

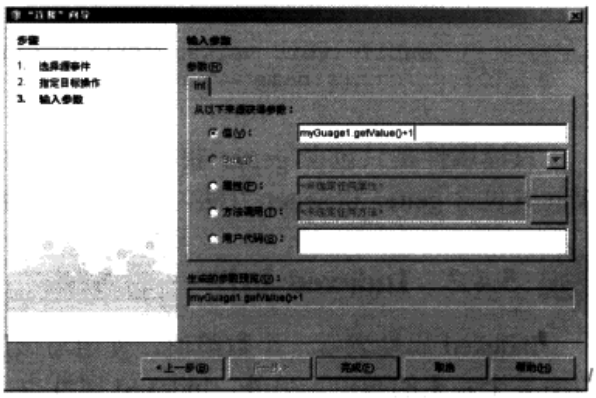


图 5-23 “输入参数”对话框

单击“完成”按钮，将自动添加如下所示的代码：

```
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

....

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    myGuage1.setValue(myGuage1.getValue()+1);
}
```

运行程序，显示界面如图 5-24 所示。当单击“更新”按钮时，进度条中间和上方会显示当前的进度值。

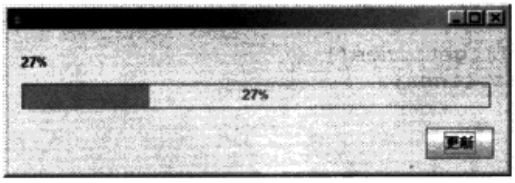


图 5-24 MyGuage 组件的应用界面

5.4 JavaBean 的属性

JavaBean 都具有属性、事件和方法三个特性，其中属性的概念相当于面向对象的程序设计语言中对象的属性概念，再说得具体些就是 JavaBean 类的成员变量。JavaBean 的属性包括 Sample 属性、Indexed 属性、Bound 属性和 Constrained 属性。

5.4.1 创建简单属性

在第 5.2.1 节创建的 MySimpleBean 组件中，name 就是一个简单类型的属性。属性一般都

Download at http://www.pin5i.com/

有与之相对应的 setter/getter 方法。例如属性 X 对应的方法为 setX/getX。其中，getX 表示读取属性的值；setX 表示设置属性的值。name 属性对应的两个 setter/getter 方法如下所示：

```
private String name;

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}
```

只读属性是只包含 getter 获取方法的属性；只写属性是只包含 setter 设置方法的属性。如果同时包含 getter 和 setter 方法，则该属性既可以被读也可以被写。



5.4.2 Indexed 属性

Indexed 属性表示一个数组值，使用与该属性对应的 setter/getter 方法可以设置或者读取数组中某个元素的数值。另外，Indexed 属性也可设置或读取整个数组的值。Indexed 属性的定义如下所示：

```
/**
 * 保存属性 lines 的值。
 */
private String[] lines;

/**
 * 属性 lines 的索引获取方法
 * @param index 属性的索引
 * @return 在 <CODE>index</CODE> 处属性的值
 */
public String getLines(int index) {
    return this.lines[index];
}

/**
 * 属性 lines 的获取方法
 * @return 属性 lines 的值
 */
public String[] getLines() {
    return this.lines;
}

/**
 * 属性 lines 的索引设置方法
 * @param index 属性的索引
 * @param lines 在 <CODE>index</CODE> 处属性的新值
 */
public void setLines(int index, String lines) {
    this.lines[index] = lines;
}

/**
 * 属性 lines 的设置方法
 * @param lines 属性 lines 的新值
 */
public void setLines(String[] lines) {
    this.lines = lines;
}
```

Download at http://www.pin5i.com/

5.4.3 Bound 属性

Bound 属性可以在 NetBeans 的属性编辑器中进行编辑，因此创建自己的组件时，一般都会使用 Bound 属性。下面的 MyJavaBean 组件就包含一个名字为 sampleProperty 的 Bound 属性。

```
import java.beans.*;
import java.io.Serializable;

/**
 * @author Liu Bin
 */
public class MyNewBean extends Object implements Serializable {

    public static final String PROP_SAMPLE_PROPERTY = "sampleProperty";

    private String sampleProperty;

    private PropertyChangeSupport propertySupport;

    public MyNewBean() {
        propertySupport = new PropertyChangeSupport(this);
    }

    public String getSampleProperty() {
        return sampleProperty;
    }

    public void setSampleProperty(String value) {
        String oldValue = sampleProperty;
        sampleProperty = value;
        propertySupport.firePropertyChange(PROP_SAMPLE_PROPERTY, oldValue,
                                           sampleProperty);
    }


    public void addPropertyChangeListener(PropertyChangeListener listener) {
        propertySupport.addPropertyChangeListener(listener);
    }

    public void removePropertyChangeListener(PropertyChangeListener listener){
        propertySupport.removePropertyChangeListener(listener);
    }

}
```

其中，PropertyChangeSupport 是一个实用工具类，支持 Bound 属性的 Bean 可以使用该类。可以使用此类的实例作为 Bean 的成员字段，并将各种工作委托给它。PropertyChangeSupport 类是可序列化的，在对它进行序列化时，它将保存（并恢复）本身可序列化的所有侦听器。在序列化期间，将跳过所有不可序列化的侦听器。

方法 firePropertyChange(PROP_SAMPLE_PROPERTY, oldValue, sampleProperty)报告所有已注册侦听器的 Bound 属性更新。如果新属性和旧属性相同并且是非空的，则不会触发事件。

方法 addPropertyChangeListener(PropertyChangeListener listener)为 JavaBean 添加一个属性监听器，当属性发生变化后，JavaBean 会通过调用 firePropertyChange 方法通知监听器的所有者。

方法 removePropertyChangeListener(PropertyChangeListener listener)用于删除已经注册的监听器。

5.5 创建简单属性编辑器

在“属性”窗口中可以直接编辑 Bean 的一些简单类型的属性，例如 String 类型。第 5.4.3

Download at http://www.pin5i.com/

节中创建的 JavaBean 就带有一个简单的属性：sampleProperty。下面在这个基础上创建一个可以添加到窗体上的 Label，并且可以在属性编辑器中修改 Label 中的文字。

1. 创建一个 Java 标准项目，项目的名字为“MyBeanPropertyEditors”。
2. 创建一个包，名字为“mybeans”。
3. 按照第 5.4.3 节中介绍的方法创建一个名为“MyLabel”的 Bean，并添加代码如下所示。

```
/*
 * MyLabel.java
 *
 * Created on 2007年2月22日, 下午9:06
 */

package mybeans;

import java.awt.Component;
import java.awt.Dimension;
import java.awt.Graphics;
import java.beans.*;
import java.io.Serializable;

/**
 * @author Liu Bin
 */
public class MyLabel extends Component implements Serializable {

    public static final String PROP_TEXT = "text";

    private String text = "MyLabel1";

    private PropertyChangeSupport propertySupport;

    public MyLabel() {
        propertySupport = new PropertyChangeSupport(this);
        this.setPreferredSize(new Dimension(60, 30));
    }

    public String getText() {
        return text;
    }

    public void setText(String value) {
        String oldValue = text;
        text = value;
        propertySupport.firePropertyChange(PROP_TEXT, oldValue, text);
    }


    public void addPropertyChangeListener(PropertyChangeListener listener) {
        propertySupport.addPropertyChangeListener(listener);
    }

    public void removePropertyChangeListener(PropertyChangeListener listener){
        propertySupport.removePropertyChangeListener(listener);
    }

    public void paint(Graphics g) {
        if (text != null && text != "") {
            g.drawString(text, 0, getHeight()-5);
        }
    }
}
```

其中，MyLabel 从 Component 派生。Component 是一个具有图形表示能力的类，可在屏幕上显示，并可与用户进行交互。典型图形用户界面中的按钮、复选框和滚动条等都是从

Download at http://www.pin5i.com/

Component 派生出来的。

在代码中实现了 void paint(Graphics g)方法，该方法用于绘制组件。这里，绘制了属性 text 中的内容。

把 MyLabel 组件添加到“组件面板”窗口中，再添加到 JFrame 窗口中，显示界面如图 5-25 所示。新添加的 text 属性在“属性”窗口中的界面如图 5-26 所示。当在“属性”窗口中修改 text 属性后，MyLabel 中的内容也会随之更新。

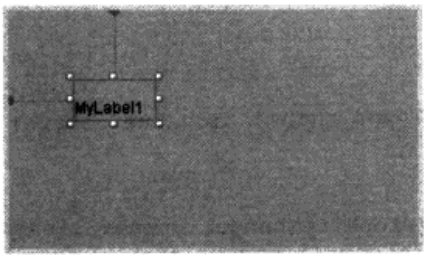


图 5-25 MyLabel 组件



图 5-26 text 属性在“属性”窗口中的界面

5.6 在“属性”窗口中实现下拉列表框

用户还可以自己定义属性编辑器，提供高级的属性编辑，例如下拉列表框形式的属性编辑器、在属性编辑器中绘制图形，甚至可以打开另外的对话框编辑属性。本节就讨论如何在“属性”窗口中实现下拉列表框。

当某个属性是枚举类型时（或者，属性值在一定的可选范围内），可以为该属性在“属性”窗口中添加下拉列表框，这样用户只能从给定的值中进行选择。下面通过一个具体的实例演示具体的实现过程。

5.6.1 创建 MyShape 组件

在第 5.5 节创建的 MyBeanPropertyEditors 标准项目中，创建一个名为“myshape”的包。在 myshape 包中创建一个 MyShape 组件，组件的代码如下所示。

```
/*
 * MyShape.java
 *
 * Created on 2007年2月22日, 下午9:16
 */

package myshape;

import java.awt.Component;
import java.awt.Dimension;
import java.awt.Graphics;
import java.beans.*;
import java.io.Serializable;
```

Download at http://www.pin5i.com/


```
/**
 * @author Liu Bin
 */
public class MyShape extends Component implements Serializable {

    public static final String PROP_SHAPE = "shape";

    private String shape = "Rectangle";

    private PropertyChangeSupport propertySupport;

    public MyShape() {
        propertySupport = new PropertyChangeSupport(this);
        this.setPreferredSize(new Dimension(50, 50));
    }

    public String getShape() {
        return shape;
    }

    public void setShape(String value) {
        String oldValue = shape;
        shape = value;
        propertySupport.firePropertyChange(PROP_SHAPE, oldValue, shape);
    }


    public void addPropertyChangeListener(PropertyChangeListener listener) {
        propertySupport.addPropertyChangeListener(listener);
    }

    public void removePropertyChangeListener(PropertyChangeListener listener){
        propertySupport.removePropertyChangeListener(listener);
    }

    public void paint(Graphics g) {
        int w = getWidth();
        int h = getHeight();
        if (shape == "Round Rectangle") {
            g.drawRoundRect(0, 0, w-1, h-1, 30, 30);
        } else if (shape == "Oval") {
            g.drawOval(0, 0, w-1, h-1);
        } else if (shape == "Arc") {
            g.drawArc(0, 0, w-1, h-1, 0, 180);
        } else {
            g.drawRect(0, 0, w-1, h-1);
        }
    }
}
```

其中，属性 shape 有如下四个可选的值。

- Rectangle：表示当前形状为一个矩形。
- Oval：表示当前形状为一个椭圆。
- Arc：表示当前形状为一个弧线。
- Round Rectangle：表示当前形状为一个圆角矩形。

paint(Graphics g)方法会根据属性 shape 的值绘制当前的图形。



5.6.2 创建下拉列表框形式的属性编辑器

myshape 包中创建一个 ShapPropertyEditor 类，用于创建下拉列表框形式的属性编辑器，代码如下所示：

```
/*
 * ShapePropertyEditor.java
```

Download at http://www.pin5i.com/


```
 *
 */

package myshape;

import java.beans.PropertyEditorSupport;

/**
 *
 * @author Liu Bin
 */
public class ShapePropertyEditor extends PropertyEditorSupport {
    public String[] getTags() {
        String shapes[] = {
            "Rectangle",
            "Round Rectangle",
            "Oval",
            "Arc"
        };
        return shapes;
    }

    public String getJavaInitializationString() {
        return "\"" + getAsText() + "\"";
    }

    public String getAsText() {
        return (String) super.getValue();
    }


    public void setAsText(String string) throws IllegalArgumentException {
        super.setValue(string);
    }
}
```

其中，PropertyEditorSupport 实现了 PropertyEditor 接口，声明如下所示：

```
public class PropertyEditorSupport extends Object implements PropertyEditor {
    ...
}
```

PropertyEditorSupport 是一个帮助构建属性编辑器的支持类，为希望允许用户编辑某个给定类型的属性值的 GUI 提供支持。PropertyEditor 提供了各种不同类型的显示和更新属性值的方式。

对于简单的属性，这里只需要实现 getAsText 和 setAsText 方法即可，其中 getAsText 方法用于返回当前属性值的字符串形式；setAsText 方法用于以字符串的形式设置属性值。因为 MyShape 的 shape 属性就是一个简单的 String 类型，因此不需要做任何的转换，这里直接调用父类的方法进行处理即可。



说明：实际上，因为 shape 属性比较简单，所以没有 getAsText 和 setAsText 方法也是可以的。对于一些复杂的属性，则需要实现这两个方法。这里给出这两个方法的实现，供读者参考。

getTags 方法用于返回属性的全部可选值，该方法返回的字符串数组将显示在“属性”窗口的属性值下拉列表框中。

getJavaInitializationString 方法将在产生代码及设置属性值时被使用。该方法需要返回一段表示当前属性值的 Java 代码。这里，在属性值字符串两边添加了引号，这样使得整个属性值会被作为一个字符串处理。

Download at http://www.pin5i.com/

5.6.3 将属性和属性编辑器关联

在创建了属性编辑器后，就需要把属性编辑器和属性关联起来。通常有两种关联的方式：通过 PropertyEditorManager 关联和通过 BeanInfo 关联，下面分别进行介绍。

1. 通过 PropertyEditorManager 进行关联

PropertyEditorManager 可用于查找任何给定类型名称的属性编辑器。此属性编辑器必须支持 java.beans.PropertyEditor 接口，以编辑给定的对象。

PropertyEditorManager 提供一个 registerEditor 方法，允许专门为某一给定类型的属性注册编辑器，其语法定义如下所示：

```
public static void registerEditor(Class<?> targetType,
                                  Class<?> editorClass)
```

参数 targetType 表示要被编辑的属性类型的 Class 对象；参数 editorClass 是编辑器类的 Class 对象。如果该参数为空，则将移除所有现有定义。例如要注册 MyString 类型属性的属性编辑器 MyStringEditor，则代码如下所示：

```
PropertyEditorManager.registerEditor(MyString.class,
                                  MyStringEditor.class);
```

使用 PropertyEditorManager 关联属性编辑器的优点是使用简单。但是缺点也是显而易见的，即不能为某一个特定的属性设置属性编辑器。

2. 通过 BeanInfo 进行关联

BeanInfo 可以对与其关联的 JavaBeans 组件的行为方式和显示方式进行说明。例如，一个 JavaBean 有多个属性和方法，那么使用 BeanInfo 可以指定哪些属性和方法可以被使用。下面介绍如何通过 BeanInfo 将 MyShape 的 shape 属性与 shape 属性编辑器 ShapPropertyEditor 进行关联。

在“项目”窗口中选中 myshape 包，然后创建一个 Java 类文件，并实现如下的代码：

```
/*
 * MyShapeBeanInfo.java
 *
 */

package myshape;

import java.beans.BeanDescriptor;
import java.beans.IntrospectionException;
import java.beans.PropertyDescriptor;
import java.beans.SimpleBeanInfo;
import mybeans.*;

/**
 *
 * @author Liu Bin
 */
public class MyShapeBeanInfo extends SimpleBeanInfo {
    private final static Class beanClass = MyShape.class;

    public BeanDescriptor getBeanDescriptor() {
        BeanDescriptor bd = new BeanDescriptor(beanClass);
        bd.setDisplayName("MyShapeBean");
        return bd;
    }

    public PropertyDescriptor[] getPropertyDescriptors() {
        try {
```

Download at http://www.pin5i.com/


```
            PropertyDescriptor textPD =
                    new PropertyDescriptor("shape", beanClass);
            textPD.setPropertyEditorClass(ShapePropertyEditor.class);
            PropertyDescriptor rv[] = {textPD};
            return rv;
        } catch (IntrospectionException e) {
            throw new Error(e.toString());
        }
    }
}
```

自己的 BeanInfo 类要从 SimpleBeanInfo 类派生。getBeanDescriptor 方法由于返回一个包含 Bean 信息的 BeanDescriptor 对象，在 BeanDescriptor 对象中可以设置 Bean 的显示名字（displayName）等信息。如果希望 Bean 信息通过自动分析获得，则方法应该返回空。

这里覆盖了 SimpleBeanInfo 的 getPropertyDescriptors 方法，用于指定哪些属性可以被显示。getPropertyDescriptors 方法返回 Bean 支持的可编辑属性的 PropertyDescriptor 数组。下面的代码创建了一个 PropertyDescriptor 对象，用于表示 MyShape 的 shape 属性。

```
PropertyDescriptor textPD =
        new PropertyDescriptor("shape", beanClass);
```

下面的语句为属性 shape 指定了属性编辑器 ShapePropertyEditor，并生成一个 PropertyDescriptor 数组。

```
textPD.setPropertyEditorClass(ShapePropertyEditor.class);
PropertyDescriptor rv[] = {textPD};
```

这样，就把属性 shape 和属性编辑器 ShapePropertyEditor 关联起来了。当把这个 Bean 添加到窗体上时，显示界面如图 5-27 所示。MyShape Bean 的 shape 属性及其属性编辑器如图 5-28 所示。

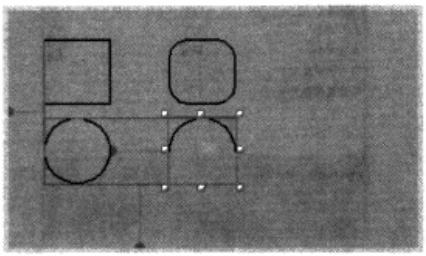


图 5-27 MyShape Bean 的显示界面

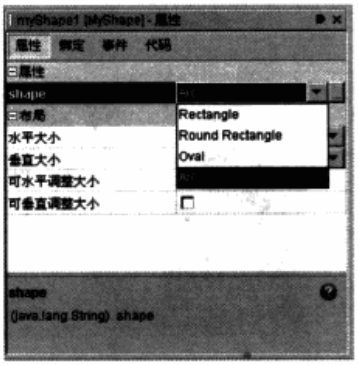


图 5-28 MyShape 的 shape 属性编辑器

5.7 绘制属性编辑器

当属性是图形时，需要在“属性”窗口中绘制属性值，例如绘制背景颜色。本节将讨论如何自己绘制属性编辑器。

PropertyEditorSupport 类的 isPaintable 方法用于决定属性编辑器是否支持自定义绘制（即是否支持 painValue 方法），其语法定义如下所示：

```
public boolean isPaintable()
```

如果希望进行自定义绘制，那么需要在派生类中覆盖这个方法，并使得方法的返回值为 true。另外，还需要覆盖 PropertyEditorSupport 类 paintValue 方法，以绘制属性。paintValue 方法的语法定义如下所示：

Download at http://www.pin5i.com/


```
public void paintValue(Graphics gfx,
                       Rectangle box)
```

paintValue 方法会在给定区域中绘制值的表示形式，参数 gfx 是要绘制的 Graphics 对象；参数 box 是应该在其中绘制图形对象的矩形。

下面把第 5.6 节中创建的 MyShape.java、ShapePropertyEditor.java 和 MyShapeBeanInfo.java 文件复制到 myshape1 包中，并在 ShapePropertyEditor 中添加如下代码：

```
public boolean isPaintable() {
    return true;
}

public void paintValue(java.awt.Graphics gfx, java.awt.Rectangle box) {
    Color oldColor = gfx.getColor();
    gfx.setColor(Color.black);
    String shape = this.getAsText();
    if (shape == "Oval") {
        gfx.drawOval(box.x, box.y, box.width-3, box.height-3);
    } else if (shape == "Round Rectangle") {
        gfx.drawRoundRect(box.x, box.y, box.width-3, box.height-3, 20, 20);
    } else if (shape == "Arc") {
        gfx.drawArc(box.x, box.y, box.width-3, box.height-3, 0, 180);
    } else {
        gfx.drawRect(box.x, box.y, box.width-3, box.height-3);
    }
    gfx.setColor(oldColor);
}
```

当把 MyShape 添加到窗体上时，会显示如图 5-29 所示的界面。对应的属性编辑器的显示界面如图 5-30 所示。可以看到，根据当前的属性值，会在“属性”窗口中绘制相应的图形。

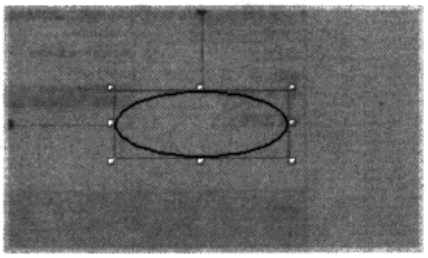


图 5-29 绘制 MyShape 的 shape 属性

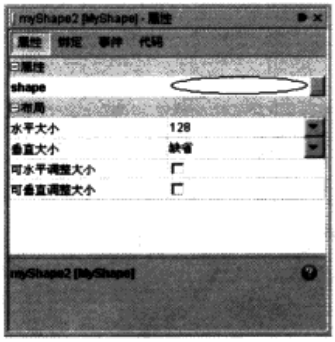


图 5-30 自绘制的属性编辑器

5.8 自定义属性编辑器

有些时候，如果属性比较复杂，可以打开一个复杂、特殊的属性编辑器，进行属性的修改。本节将讨论如何设计自定义属性编辑器。

PropertyEditorSupport 类的 supportsCustomEditor 方法用于确定此属性编辑器是否支持自定义编辑器。如果提供定义的属性编辑器，则需要在派生类中覆盖这个方法，并使得方法的返回值为 true。

另外，还需要覆盖 getCustomEditor 方法，返回自定义的属性编辑器。getCustomEditor 方法的语法定义如下所示：

```
Component getCustomEditor()
```

通常会先创建一个可以编辑属性的窗体，然后在这个方法中创建这个窗体，并通过方法返回。

Download at http://www.pin5i.com/

下面把第 5.6 节中创建的 MyShape.java、ShapePropertyEditor.java 和 MyShapeBeanInfo.java 文件复制到 myshape2 包中，并在 ShapePropertyEditor 中添加如下代码：

```
public boolean supportsCustomEditor() {
    return true;
}

public Component getCustomEditor() {
    return new CustomShape(this);
}
```

其中 CustomShape 为自定义的属性编辑器类。该类继承于 JPanel 类。在面板上添加如图 5-31 所示的组件，并编写如下代码。

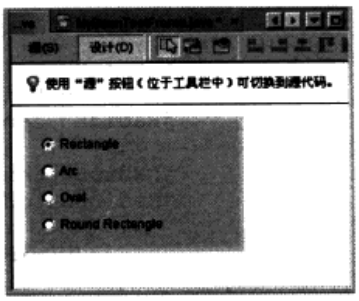


图 5-31 自定义的属性编辑器界面

```
/*
 * CustomShape.java
 *
 */

package myshape2;

import java.beans.Customizer;
import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeSupport;
import java.util.Enumeration;
import javax.swing.JRadioButton;

/**
 *
 * @author  Liu Bin
 */
public class CustomShape extends javax.swing.JPanel {
    private ShapePropertyEditor editor;

    public CustomShape(ShapePropertyEditor editor) {
        this.editor = editor;
        initComponents();
        setSelectedStatus();
    }

    // <editor-fold defaultstate="collapsed" desc=" 生成的代码 ">
    private void initComponents() {
        buttonGroup1 = new javax.swing.ButtonGroup();
        jRadioButton1 = new javax.swing.JRadioButton();
        jRadioButton2 = new javax.swing.JRadioButton();
        jRadioButton3 = new javax.swing.JRadioButton();
        jRadioButton4 = new javax.swing.JRadioButton();

        buttonGroup1.add(jRadioButton1);
        jRadioButton1.setSelected(true);
        jRadioButton1.setText("Rectangle");
        jRadioButton1.setBorder(javax.swing.BorderFactory.createEmptyBorder(0,
0, 0, 0));
        jRadioButton1.setMargin(new java.awt.Insets(0, 0, 0, 0));
        jRadioButton1.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                jRadioButtonItemStateChanged(evt);
            }
        });

        buttonGroup1.add(jRadioButton2);
        jRadioButton2.setText("Arc");
        jRadioButton2.setBorder(javax.swing.BorderFactory.createEmptyBorder(0,
0, 0, 0));
        jRadioButton2.setMargin(new java.awt.Insets(0, 0, 0, 0));
        jRadioButton2.addItemListener(new java.awt.event.ItemListener() {
```

Download at http://www.pin5i.com/


```
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                jRadioButtonItemStateChanged(evt);
            }
        });

        buttonGroup1.add(jRadioButton3);
        jRadioButton3.setText("Oval");
        jRadioButton3.setBorder(javax.swing.BorderFactory.createEmptyBorder(0,
0, 0, 0));
        jRadioButton3.setMargin(new java.awt.Insets(0, 0, 0, 0));
        jRadioButton3.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                jRadioButtonItemStateChanged(evt);
            }
        });

        buttonGroup1.add(jRadioButton4);
        jRadioButton4.setText("Round Rectangle");
        jRadioButton4.setBorder(javax.swing.BorderFactory.createEmptyBorder(0,
0, 0, 0));
        jRadioButton4.setMargin(new java.awt.Insets(0, 0, 0, 0));
        jRadioButton4.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                jRadioButtonItemStateChanged(evt);
            }
        });

        org.jdesktop.layout.GroupLayout layout = new
org.jdesktop.layout.GroupLayout(this);
        this.setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
            .add(layout.createSequentialGroup()
            .addContainerGap()
            .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                    .add(jRadioButton1)
                    .add(jRadioButton2)
                    .add(jRadioButton3)
                    .add(jRadioButton4))
                .addContainerGap(72, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
        layout.createParallelGroup(org.jdesktop.layout.GroupLayout. LEADING)
            .add(layout.createSequentialGroup()
                .addContainerGap()
                .add(jRadioButton1)
                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                .add(jRadioButton2)
                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                .add(jRadioButton3)
                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                .add(jRadioButton4)
                .addContainerGap(org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            );
    }// </editor-fold>

    private void jRadioButtonItemStateChanged(java.awt.event.ItemEvent evt) {
        updateSelectedStatus();
        editor.firePropertyChange();
    }


    // 变量声明 - 不进行修改
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
```

Download at http://www.pin5i.com/


```
    private javax.swing.JRadioButton jRadioButton3;
    private javax.swing.JRadioButton jRadioButton4;
    // 变量声明结束

    public void setSelectedStatus() {
        for (Enumeration e=buttonGroup1.getElements();e.hasMoreElements();) {
            JRadioButton rb = (JRadioButton)e.nextElement();
            if (rb.getText().equalsIgnoreCase(editor.getAsText())) {
                rb.setSelected(true);
                break;
            }
        }
    }

    public void updateSelectedStatus() {
        for (Enumeration e=buttonGroup1.getElements();e.hasMoreElements();) {
            JRadioButton rb = (JRadioButton)e.nextElement();
            if (rb.isSelected()) {
                editor.setValue(rb.getText());
            }
        }
    }

}
```

当把 MyShape 添加到窗体上，并在“属性”窗口中单击 shape 属性旁边的“...”按钮时，会打开如图 5-32 所示的属性编辑界面。

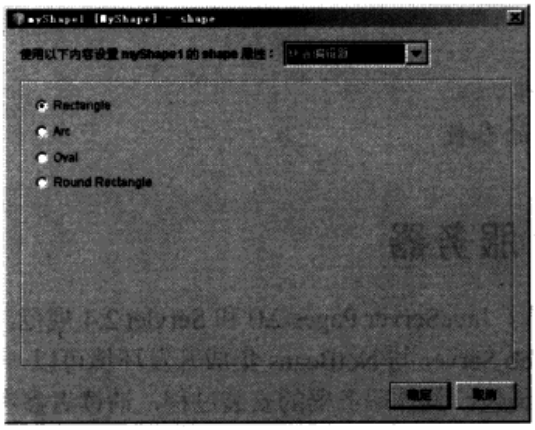


图 5-32 自定义的属性编辑器

5.9 小结

JavaBean 是 Java 技术中很重要的一个技术，通过 JavaBean 技术可以实现代码的复用。本章主要介绍了在 NetBeans 中快速开发 JavaBean 的各种方法，包括如何通过模板创建一个简单的 JavaBean、如何创建 JavaBean 的属性、如何定制属性编辑器，以及如何生成 BeanInfo 等内容。

另外，值得一提的是，本章中还给出了通过连接模式实现事件连接的方法。通过连接模式可以快速产生代码，因此读者应该好好掌握这种方法。

Download at http://www.pin5i.com/

第 6 章

管理和使用服务器

服务器是供应用程序运行的计算机。通过安装服务器软件并将计算机连接至互联网，可以使任何计算机成为服务器。在 NetBeans 集成开发环境中支持多个服务器，例如 Java EE 服务器、Web 服务器等。本章将讨论如何在 NetBeans 中注册和管理一些常用的服务器，例如 GlassFish、Sun Java Application Server、Tomcat 等。

本章主要内容

- 启动和停止服务器
- 注册各个不同服务器的方法
- 服务器的管理
- 配置服务器的参数

本章亮点

- 注册 Tomcat 服务器
- GlassFish 的配置管理
- 设置 Tomcat 服务器的参数

6.1 Tomcat 服务器

Tomcat Web Server 实现了 JavaServer Pages 2.0 和 Servlet 2.4 规范，并且包含许多用于部署 Web 应用程序的功能。Tomcat Web Server 与 NetBeans 集成开发环境可以一起安装，要使用该服务器，需要下载包含全部功能的安装包。具体服务器的安装过程，请读者参考第 1 章中的安装过程。

6.1.1 启动和停止服务器

在“运行环境”窗口中展开“服务器”节点，会显示 NetBeans 集成开发环境捆绑的 Tomcat 服务器，如图 6-1 所示。

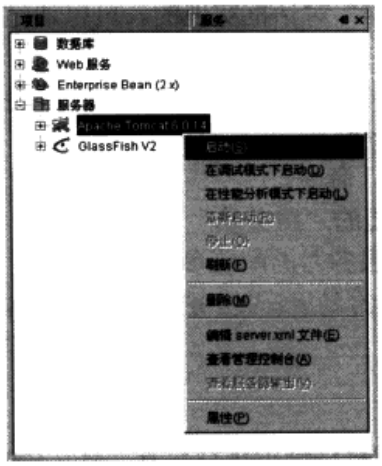
右键单击“Apache Tomcat 6.0.14”节点，选择弹出菜单中的“启动”命令，可以启动 Tomcat 服务器。启动 Tomcat 后，“运行环境”窗口的显示界面如图 6-2 所示。

在“输出”窗口中会显示 Tomcat 启动过程的信息，如图 6-3 所示。选择 Tomcat 节点弹出菜单中的“查看服务器输出”命令，也可以在“输出”窗口中显示 Tomcat 的输出信息。

单击 按钮可以重新启动服务器。单击 按钮会停止服务器的运行。单击 按钮会刷新服务器的状态。当 Tomcat 服务器停止时，单击 按钮会再次启动 Tomcat 服务器，或者单击 按钮会在调试模式下启动 Tomcat 服务器。

另外，在“Apache Tomcat 6.0.14”节点的弹出菜单中，也可以选择“在调试模式下启动”命令，以调试模式启动 Tomcat。选择“重新启动”命令，以重新启动 Tomcat。选择“停止”命令，以停止 Tomcat 服务器的运行。选择“刷新”命令，以刷新服务器的状态。

Download at http://www.pin5i.com/



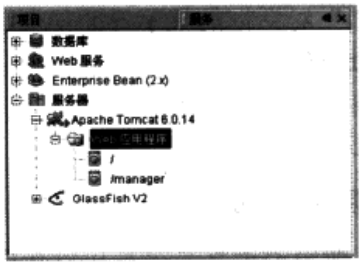


图 6-1 NetBeans 集成开发环境捆绑的 Tomcat 服务器 图 6-2 启动 Tomcat 后的“运行环境”窗口

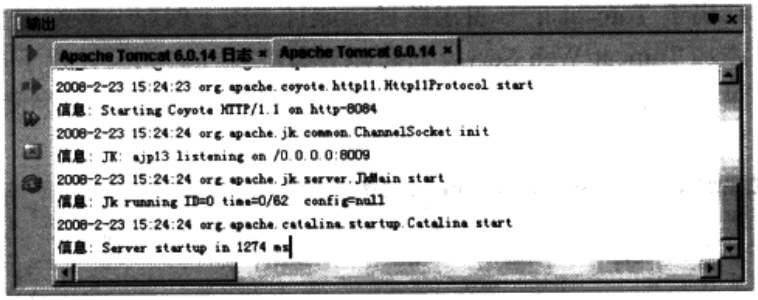


图 6-3 “输出”窗口中显示 Tomcat 启动过程的信息

6.1.2 编辑 Tomcat 配置文件

Tomcat 的主配置文件是 server.xml 文件，它提供了 Tomcat 组件的配置信息，并指定了服务器的部署信息。对于简单的 Web 应用程序，因为集成开发环境会自动进行必要的更改，所以不需要编辑 server.xml 文件。只有对于高级配置才需要进行编辑，如基于 JNDI 或数据库查找的验证。

server.xml 文件通常位于 Tomcat 安装目录的 conf 子目录中。也可以通过以下的方法在源代码编辑器中打开 server.xml 文件进行编辑。

❶ 在“运行环境”窗口中，展开“服务器”→“Apache Tomcat 6.0.14”节点。

❷ 如果 Tomcat 服务器已经在运行，那么右键单击“Apache Tomcat 6.0.14”节点，选择弹出菜单中的“停止”命令，停止服务器。

❸ 右键单击“Apache Tomcat 6.0.14”节点，选择弹出菜单中的“编辑 server.xml”命令，打开 server.xml 文件进行编辑。

❹ 重新启动 Tomcat，新的设置就会生效。

注意：手动编辑 server.xml 文件有一定的风险，因为集成开发环境无法修复已损坏的 server.xml 文件。强烈建议在开始手动编辑之前，对 server.xml 文件进行备份。

6.1.3 定制 Tomcat 服务器

1. 打开“服务器”对话框

在 NetBeans 集成开发环境中可以定制 Tomcat 服务器的一些基本设置，例如修改服务器的

Download at http://www.pin5i.com/

端口号等。定制 Tomcat 服务器是通过“服务器管理器”对话框实现的。可以通过下面的两种方式打开该对话框。

❶ 在“运行环境”窗口中展开树视图中的“服务器”节点和 Tomcat Web 服务器节点。选择 Tomcat Web 服务器节点上下文菜单中的“属性”命令。

❷ 选择 NetBeans 主菜单的“工具”→“服务器”命令。

“服务器”对话框的显示界面如图 6-4 所示。左边的“服务器”列表中列出了所有支持的服务器。当选择某个服务器后，会在右边部分显示该服务器的配置参数。

“服务器名称”文本框中，显示了 Tomcat 服务器的名字。“服务器类型”文本框中显示的是服务器的类型。

2. 设置连接参数

Tomcat 服务器的连接参数可以在如图 6-4 所示的对话框中进行设置。在“Catalina Home 目录”文本框中，给出了 Tomcat 服务器的实际路径，通常又称作 Home 目录。在“Catalina 基目录”文本框中，给出了 Tomcat 的基目录路径，该目录用于保存一些配置文件。向 NetBeans 的服务器管理器中添加 Tomcat Web Server 时可以设置此属性。如果设置了基目录，Tomcat 配置文件将放在该目录中。指定单独基目录的原因之一是为了允许多个用户使用同一个服务器。如果没有指定基目录，配置文件将位于 Home 目录中。

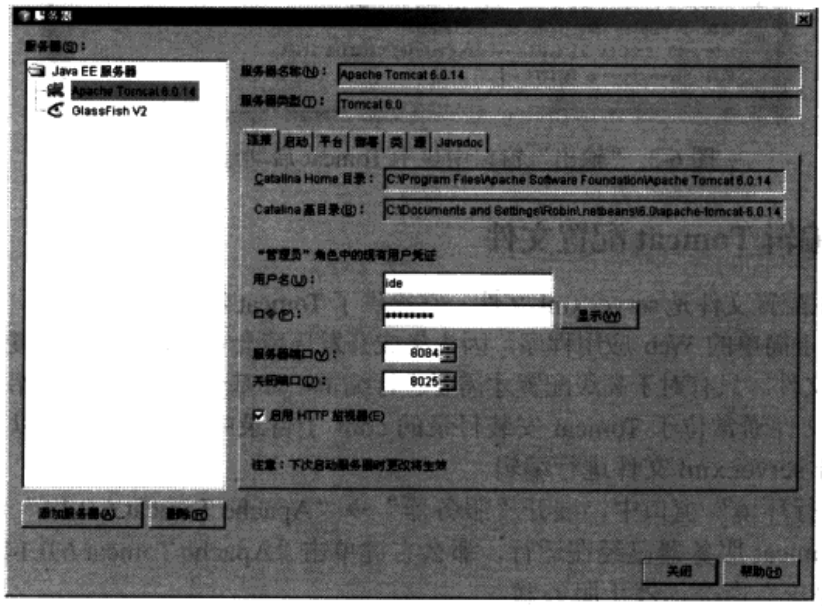


图 6-4 Tomcat Web 服务器属性的“连接”标签

在“用户名”文本框中，可以设置管理员角色的用户名。在“口令”文本框中可以修改管理员的密码。单击“显示”按钮，可以显示当前的密码。

“服务器端口”文本框用于修改服务器使用的端口。“关闭端口”文本框用于设置关闭服务器时使用的端口。

3. 设置启动参数

Tomcat Web 服务器属性的“启动”标签如图 6-5 所示。如果选中“使用定制的 Catalina 脚本”复选框，那么可以在下面的“Cataline 脚本”文本框中输入自己的脚本文件。在“共享内存的名称”文本框中，可以修改为其他的名字。在“套接字端口”微调文本框中可以修改 Socket 的端口号。

Download at http://www.pin5i.com/

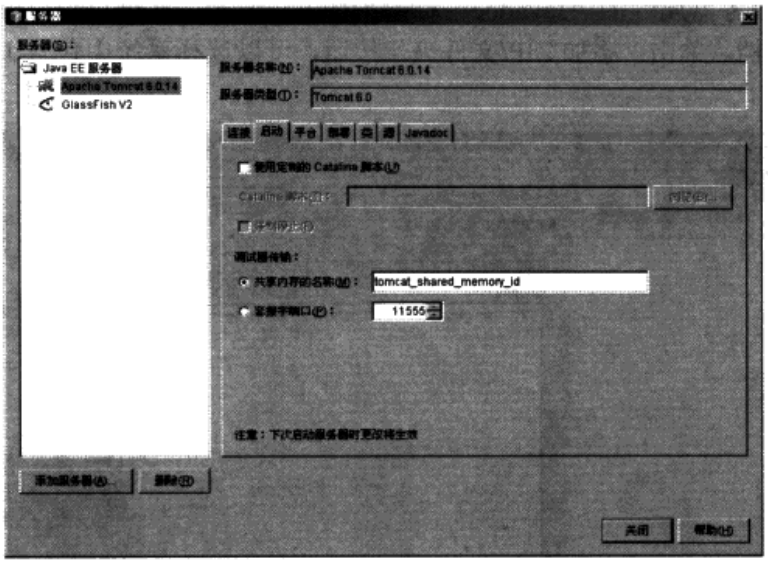


图 6-5 Tomcat Web 服务器属性的“启动”标签

4. 设置平台参数

Tomcat Web 服务器属性的“平台”标签如图 6-6 所示。在这里可以选择 Tomcat Web 服务器使用的平台。在“Java 平台”下拉列表框中，可以选择使用其他的 Java JDK 版本（如果系统中安装了多个 JDK 的话）。

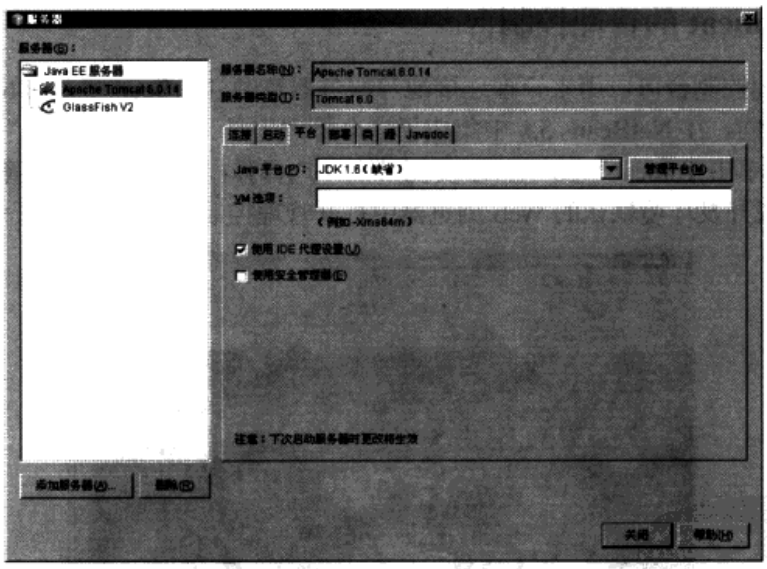


图 6-6 Tomcat Web 服务器属性的“平台”标签

“VM 选项”文本框中可以输入虚拟机的参数，一般情况下不需要指定任何参数。选中“使用 IDE 代理设置”后，在需要网络连接的时候，可以使用 NetBeans 集成开发环境的代理服务器设置。如果选中“使用安全管理器”复选框，可以使用安全管理器来管理 MIDP 项目的安全密钥库。

5. 其他参数设置

Tomcat Web 服务器属性的“类”标签中列出了所有使用的JAR 文件及其路径。如果要添加其他的库，可以选择“源”标签，再单击“添加 JAR/文件夹”命令进行添加。

Download at http://www.pin5i.com/

Tomcat Web 服务器属性的“Javadoc”标签如图 6-7 所示。在这里可以添加 Tomcat 服务器使用的 Javadoc 文档。单击“添加 ZIP/文件夹”按钮，可以选择系统中包含 Javadoc 文档的 ZIP 文件或者目录。

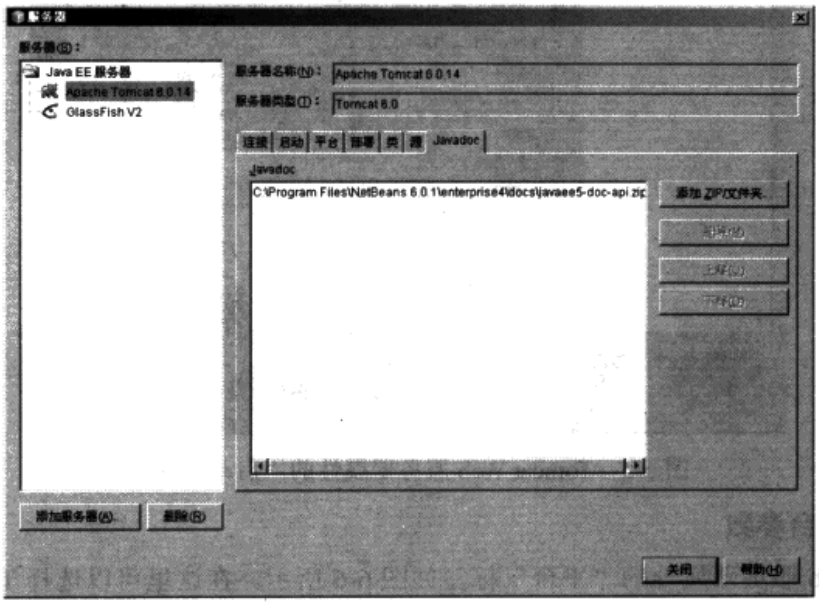


图 6-7 Tomcat Web 服务器属性的“Javadoc”标签

6.1.4 Tomcat 的管理控制台

Tomcat 的管理控制台可以用来管理连接池等服务器资源。NetBeans 5.x 和 NetBeans 6.0 对这个工具的支持有所不同。在 NetBeans 5.x 中集成了这个管理工具。在“运行环境”窗口中，展开“服务器”节点，右键单击 Tomcat Web Server 实例节点，选择弹出菜单的“查看管理控制台”命令，会在 NetBeans 集成开发环境默认的 Web 浏览器中打开管理控制台的登录界面，如图 6-8 所示。



图 6-8 Tomcat 管理控制台的登录界面

默认情况下，管理员的账号为 ide，密码为 G2bwGm8b。如果需要修改账号或者密码，可以打开 tomcat-users.xml 文件进行修改。tomcat-users.xml 文件位于 Tomcat 基目录（具体可以参考图 6-4 中的 Catalina 基目录设置）的 conf 文件夹中。tomcat-users.xml 文件的内容如下所示。

Download at http://www.pin5i.com/


```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <role rolename="manager"/>
  <role rolename="admin"/>
  <user username="ide" password="G2bwGm8b" roles="manager,admin"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="role1" password="tomcat" roles="role1"/>
  <user username="both" password="tomcat" roles="tomcat,role1"/>
</tomcat-users>
```

在 tomcat-users.xml 文件中，检查是否存在名为“admin”的角色，以及是否已将该角色分配给用户。如果需要，请自己定义 admin 角色，并将该角色分配给用户。修改完毕后，需要重新启动 Tomcat 服务器才能生效。

在输入正确的用户名和密码后，单击 Login 按钮，即可登录到 Tomcat 管理控制台。在这里可以查看 Tomcat 服务器的资源和服务，以及设置用户、分组、角色等。对于某个用户，还可以修改用户密码等。

在 NetBeans 6.0 中，如果运行该命令，则会显示如图 6-9 所示的界面。这是因为该管理工具已经不包含在 NetBeans 集成开发环境中了，如果需要使用管理功能，需要单独下载安装 Tomcat 管理包。

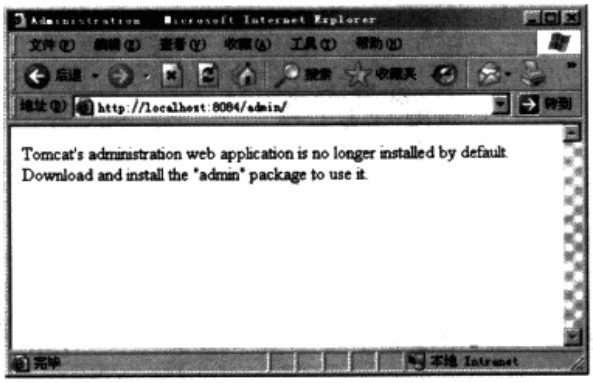


图 6-9 Tomcat 管理控制台

6.1.5 注册外部的 Tomcat 服务器

在 NetBeans 中，除了可以使用内置的 Tomcat 服务器外，还可以添加外部的 Tomcat 服务器。当需要使用某个特定版本的 Tomcat 服务器时，这个功能就显得十分有用了。注册外部的 Tomcat 服务器的步骤介绍如下。

❶ 选择 NetBeans 主菜单的“工具”→“服务器”命令，打开“服务器管理器”对话框。单击“添加服务器”按钮，打开“选择服务器”对话框，如图 6-10 所示。

❷ 在“服务器”列表中选择 Tomcat 服务器的类型，例如这里选择 Tomcat 5.5 服务器。单击“下一步”按钮，打开“安装和登录详细信息”对话框，如图 6-11 所示。

❸ 在“Catalina Home 目录”文本框中输入外部 Tomcat Web 服务器的目录。如果希望为不同的用户保留不同的 Tomcat 配置文件，则需要选中“使用专用配置文件夹（Catalina 基文件夹）”复选框，然后在“Catalina 基目录”文本框中输入保存配置文件的基目录。在“用户名”文本框中输入管理 Tomcat 服务器的管理员账号，在“口令”文本框中输入管理员的密码。单击“完成”按钮，完成外部 Tomcat 服务器的注册。注册完毕后，可以在“运行环境”窗口中看到添加的 Tomcat 服务器，如图 6-12 所示。

Download at http://www.pin5i.com/

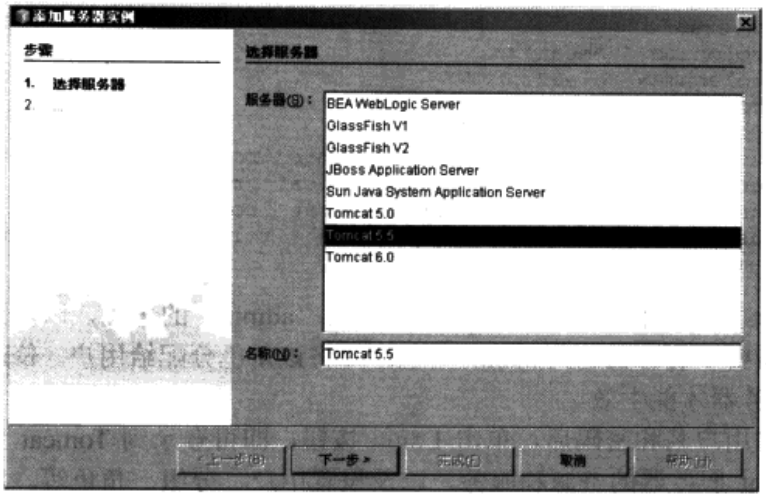


图 6-10 “选择服务器”对话框

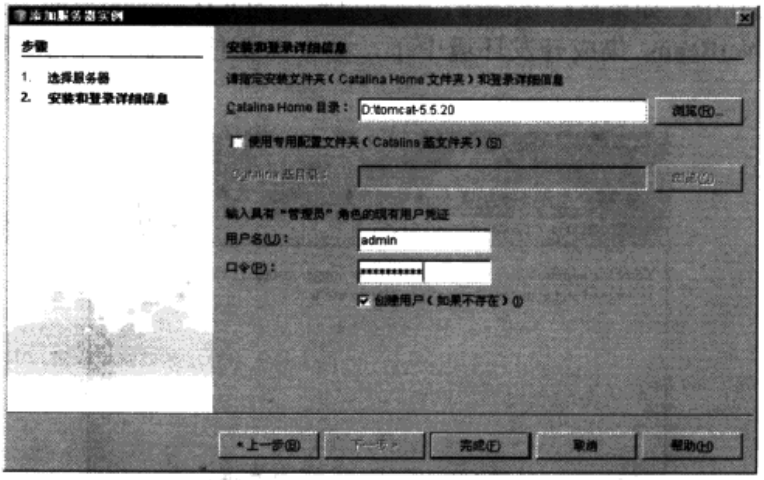


图 6-11 “安装和登录详细信息”对话框

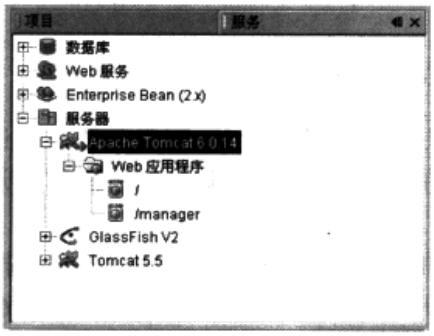


图 6-12 已经注册的外部 Tomcat 服务器

6.2 GlassFish 和 Sun Java 应用程序服务器

在 NetBeans 5.x 中一般使用 Sun Java 应用程序服务器（Sun Java System Application Server）进行 J2EE 的开发。在 NetBeans 6.0 中默认的应用程序服务器变为 GlassFish。

GlassFish 是用于构建 Java EE 5 应用服务器的开源开发项目的名称。它基于 Sun 公司提供的 Sun Java System Application Server PE 9 的源代码以及 Oracle 公司贡献的 TopLink 持久性代码。

Download at http://www.pin5i.com/

GlassFish 项目提供了开发高质量应用服务器的结构化过程，以前所未有的速度提供新的功能。

由于 GlassFish 已经默认的安装在系统中，因此下面就以 Sun Java 应用程序服务器为例介绍独立的应用程序服务器的安装，以及如何在 NetBeans 中进行注册和管理。

6.2.1 安装 Sun Java 应用程序服务器

Sun Java 应用程序服务器是与 Java EE 兼容的应用服务器，它支持基于 Java Server Pages（JSP）、Java Servlet 和 Enterprise JavaBeans（EJB）组件的应用程序。

可以使用 Java System Application Server 包含的库来编译 Web 服务和客户端，验证应用程序是否正确实现了 Java EE 规范，创建具有 Java Server Faces（JSF）支持的新 Web 应用程序，或者将 JSF 支持添加到现有的 Web 应用程序中。

Sun Java 应用程序服务器及其库可以执行下列操作：

- 部署企业应用程序和 Enterprise Bean；
- 部署 Web 服务和使用 JSR-109 或 JSR-101 桩模块的 Web 服务客户端；
- 部署 Web 应用程序、JSP 页面和 Servlet。

在 Sun 的官方网站上下载 Sun Java System Application Server，双击安装文件开始安装，显示如图 6-13 所示的欢迎界面。

单击“下一步”按钮，打开“软件许可协议”窗口，如图 6-14 所示。

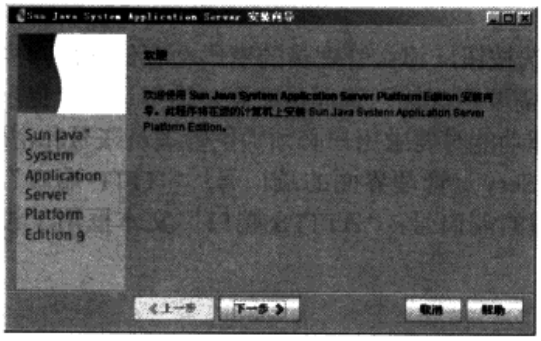


图 6-13 欢迎界面

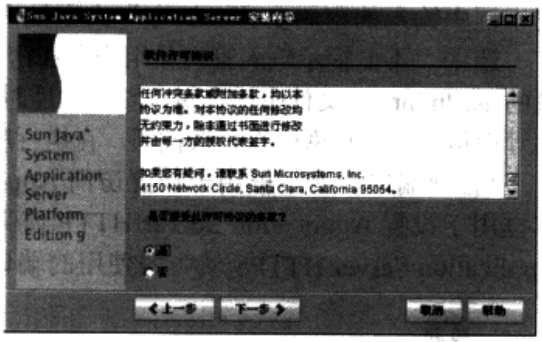


图 6-14 “软件许可协议”窗口

选择“是”单选按钮，单击“下一步”按钮，打开“选择安装目录”窗口，如图 6-15 所示。

可以在“安装目录”文本框中输入服务器的安装路径，或者单击“浏览”按钮进行选择。这里使用默认的路径，单击“下一步”按钮。如果目录不存在，会显示“是否要创建新目录”对话框，如图 6-16 所示。

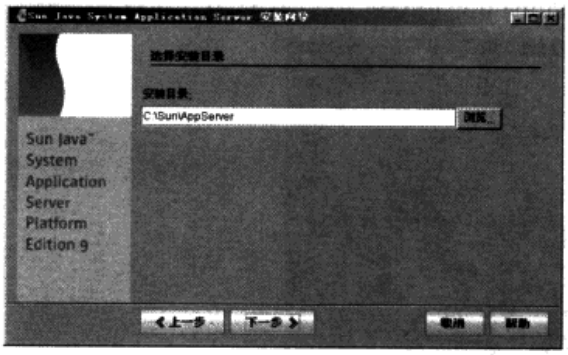


图 6-15 “选择安装目录”窗口



图 6-16 “是否要创建新目录”对话框

单击“创建目录”按钮会自动创建目录，单击“选择新目录”按钮会返回“选择安装目录”

Download at http://www.pin5i.com/

窗口，要求用户重新进行选择。这里，单击“创建目录”按钮，打开“需要 JDK”窗口，如图 6-17 所示。

在“JDK 位置”文本框中输入 JDK 的路径，或者单击“浏览”按钮进行选择。单击“下一步”按钮，打开“管理配置”窗口，如图 6-18 所示。

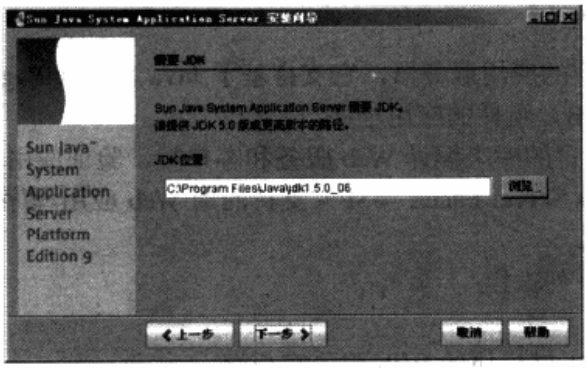


图 6-17 “需要 JDK”窗口

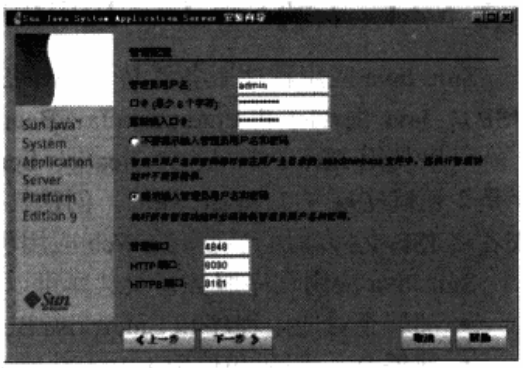


图 6-18 “管理配置”窗口

“管理配置”窗口用于输入基本配置参数。“管理员用户名”文本框是管理此域的默认管理员用户的名称（字母数字字符）。

“口令”文本框用于输入管理员用户的密码，密码长度至少为 8 个字符。“重新输入口令”文本框中输入的密码要和“口令”文本框中的一样。

选中“不要提示输入管理员用户名和密码”单选按钮，那么管理员的密码会存储在用户主目录的.asadminpass 文件中，在执行管理功能时用户不需要再次输入管理员密码。这里选中“提示输入管理员用户名和密码”单选按钮，在执行所有管理功能时要求用户必须提供管理员账号和密码。

“管理端口”文本框用于设置访问 Application Server 管理界面的端口号；“HTTP 端口”文本框用于设置 Application Server HTTP 侦听器使用的端口号；“HTTPS 端口”文本框用于设置 Application Server HTTPS 侦听器使用的端口号。



说明：端口号值必须是数字，其有效值介于 1 和 65535 之间。对于 UNIX 系统，如果用户为非超级用户，那么使用 1024 和 65535 之间的值。选定的端口不能是已被占用的端口。如果选定的端口已被占用，安装仍然可以继续，但是安装完毕后不一定能够启动服务器（因为冲突）。当然，如果能在尝试启动服务器之前使这些端口可用，这也是可以的。

单击“下一步”按钮，打开“安装选项”窗口，如图 6-19 所示。如果选中“注册 Application Server”单选按钮，那么在安装完毕 Sun Java Application Server 后，会显示一个注册页面。

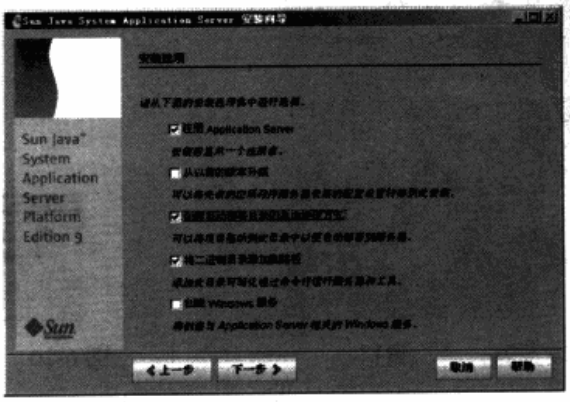


图 6-19 “安装选项”窗口

Download at http://www.pin5i.com/

选中“从以前的版本升级”前的复选框，可以将以前的 Java 应用程序服务器安装的配置转移到此安装。选中“创建自动部署目录的桌面快捷方式”前的复选框，将在桌面上创建一个自动部署目录的快捷方式。用户可以通过这个快捷方式，迅速地部署应用程序。

选中“将二进制目录添加到路径”前的复选框，会把包含可执行文件的路径添加到 PATH 中，这样可以快速地执行服务器和工具。选中“创建 Windows 服务”前的复选框会为应用程序服务器创建一个 Windows 服务，这样可以选择是否随着 Windows 的启动而自动启动应用程序服务器。

单击“下一步”按钮，打开“准备安装”窗口，显示程序的安装信息等内容，如图 6-20 所示。

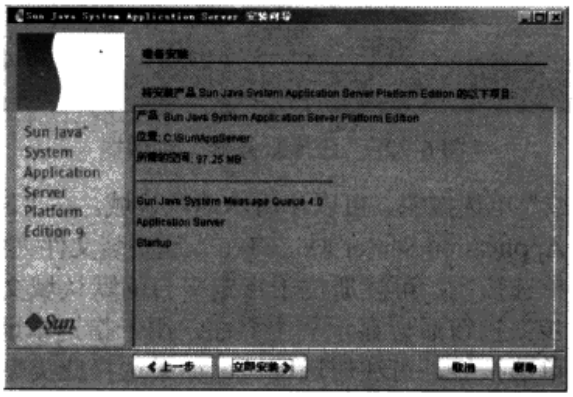


图 6-20 “准备安装”窗口

单击“立即安装”按钮开始安装程序，安装完毕后显示如图 6-21 所示的“安装完成”窗口。此时，如果单击“启动服务器”按钮，则会启动应用程序服务器。单击“完成”按钮会完成服务器程序的安装。

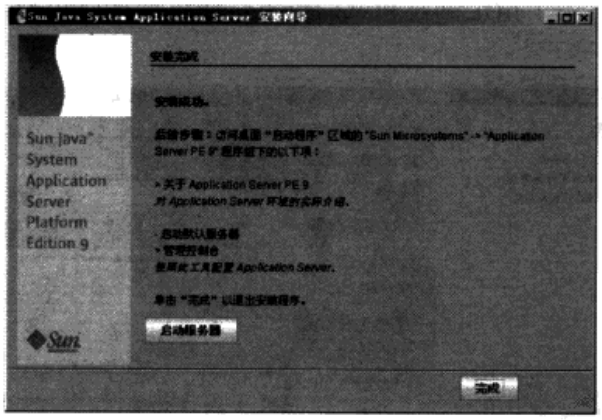


图 6-21 “安装完成”窗口

6.2.2 注册 Sun Java 应用程序服务器

在 Sun Java 应用程序服务器安装后，需要在 NetBeans 集成开发环境进行注册，这样才能够使用这个服务器。

选择 NetBeans 主菜单中的“工具”→“服务器”命令，打开“服务器”对话框。单击“添加服务器”按钮，打开“选择服务器”对话框。在“服务器”列表中选择 Sun Java System Application Server，如图 6-22 所示。

单击“下一步”按钮，打开“平台文件夹位置”对话框，如图 6-23 所示。在“平台位置”文本框中输入 Sun Java Application Server 的本地安装路径。也可以单击“浏览”按钮选择该位置。

Download at http://www.pin5i.com/

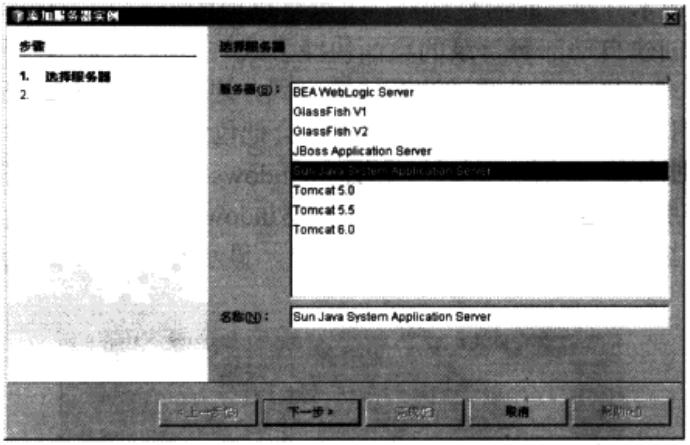


图 6-22 “选择服务器”对话框

选中“注册本地默认域”单选按钮，可以注册本地默认域。在“域”下拉列表框中选择一个域，安装 Sun Java System Application Server 时，将在本地安装文件夹中创建包含默认域文件夹。

选中“注册本地域”单选按钮，可注册位于指定平台的默认域文件夹以外的本地域。如果选择此类型，单击“下一步”按钮后会显示一个界面，用于指定本地域的位置。

选中“注册远程域”单选按钮，可注册远程域。如果选择此类型，单击“下一步”按钮后会显示一个界面，用于指定远程域的主机和通信端口。



注意： *无法从 NetBeans 集成开发环境中停止或启动远程服务器。*

选中“创建个人域”单选按钮，可以创建一个新的域。如果选择此类型，单击“下一步”按钮后会显示一个界面，用于指定创建新域的位置。通过选择此类型，开发人员可以为新域指定端口设置。

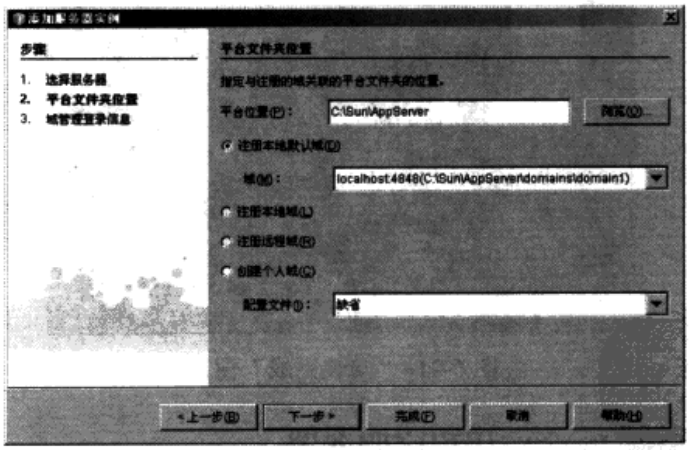


图 6-23 “平台文件夹位置”对话框

这里，选中“注册本地默认域”单选按钮。单击“下一步”按钮，打开“域管理登录信息”对话框，用于指定作为域管理员登录时使用的用户名和口令，如图 6-24 所示。

在“管理员用户名”文本框中可以输入域管理员的用户名；在“管理员口令”文本框中可以输入域管理员的口令。

注册 Sun Java Application Server 时，用户无须提供域管理员用户名和口令。如果未在注册实例时提供用户名和口令，则在需要时仍然会提示用户提供用户名和口令。

Download at http://www.pin5i.com/

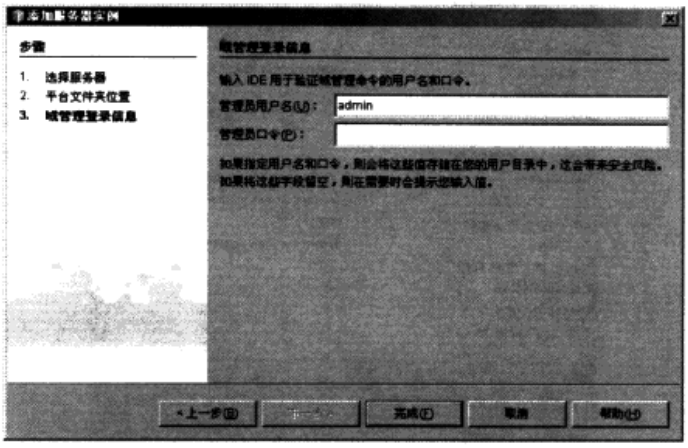


图 6-24 “域管理登录信息”对话框

提醒：如果输入用户名和口令，则会将这些值存储在用户目录中。这可能会带来安全风险。

单击“完成”按钮，完成 Sun Java Application Server 的注册。在 NetBeans 集成开发环境中启动 Sun Java Application Server，然后在“服务”窗口中展开 Sun Java System Application Server 节点，如图 6-25 所示。

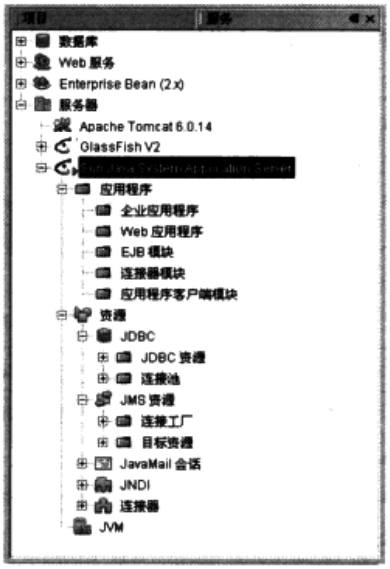


图 6-25 Sun Java System Application Server 节点

6.2.3 修改 Sun Java Application Server 的 JVM 属性

通过在“运行环境”窗口中修改 JVM 属性表单，可以设置和修改 Sun Java System Application Server 的 JVM 属性。

在“运行环境”窗口中，展开“服务器”节点和 Sun Java System Application Server 实例。右键单击 JVM 节点，然后选择弹出菜单中的“属性”命令，打开“JVM - 属性”窗口，如图 6-26 所示。在属性窗口中，各个属性的说明如表 6-1 所示。

Download at http://www.pin5i.com/

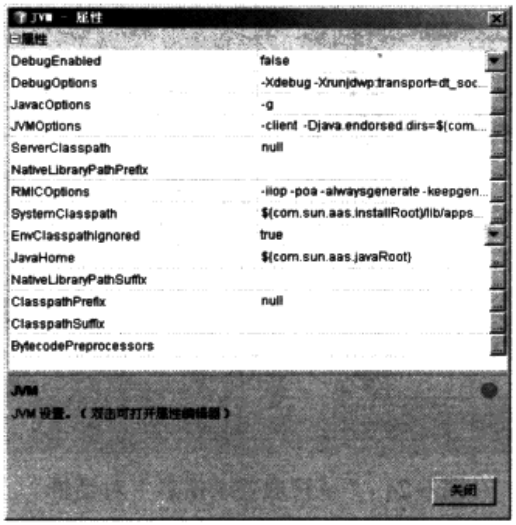


图 6-26 “JVM - 属性”窗口

表 6-1 “JVM - 属性”窗口中属性的说明

属　性	说　明
BytecodePreprocessors	输入以逗号分隔的类名列表。每个类都必须实现 com.sun.appserv.BytecodePreprocessor 接口。将按指定的顺序调用类
JavacOptions	显示 javac 选项。直接编辑选项，或单击该框以使用编辑器来编辑选项
DebugEnabled	设置为 true 可启用调试模式；设置为 false 可禁用调试模式。如果设置为 true，则设置 jpda_port_number
SystemClasspath	修改系统类路径
ServerClasspath	单击该框可在服务器类路径中添加目录或 JAR/ZIP 文件。也可以删除当前位于服务器类路径中的目录或 JAR/ZIP 文件，并重新排列类路径引用顺序
NativeLibraryPathPrefix	在本地库路径的开头添加条目。本地库路径是由服务器的本地共享库相对路径、标准 JRE 本地库路径、shell 环境设置（UNIX 上为 LD_LIBRARY_PATH）以及“JVM Profiler 设置”页中指定的任何路径连接而成的一个串
DebugOptions	输入或编辑调试选项。请注意，仅在 Windows 上支持使用 transport=dt_shmem 选项
JVMOptions	显示 JVM 选项。直接编辑选项，或单击该框以使用编辑器来编辑选项
JavaHome	Java SDK 的安装目录名
NativeLibraryPathSuffix	在本地库路径的末尾添加条目。本地库路径是由服务器的本地共享库相对路径、标准 JRE 本地库路径、shell 环境设置（UNIX 上为 LD_LIBRARY_PATH）以及“JVM Profiler 设置”页中指定的任何路径连接而成的一个串
ClasspathSuffix	单击该框可在类路径的末尾添加目录或 JAR/ZIP 文件。也可以删除当前位于 JVM 类路径中的目录或 JAR/ZIP 文件，并重新排列类路径引用顺序
ClasspathPrefix	单击该框可在类路径的开头添加目录或 JAR/ZIP 文件。也可以删除当前位于类路径中的目录或 JAR/ZIP 文件，并重新排列类路径引用顺序
EnvClasspathIgnored	如果设置为 true，则忽略 EnvClasspath 的设置
RMICOptions	RMIC 编译器的命令行选项



6.2.4 管理 GlassFish & Sun Java Application Server

GlassFish 管理界面和 Sun Java Application Server 的管理界面基本一致，因此这里仍然以

Download at http://www.pin5i.com/

Sun Java System Application Server 管理控制台为介绍目标。Application Server 管理控制台（简称为管理控制台）是一种基于浏览器的工具，用于管理域管理服务器（Domain Administration Server, DAS）。



说明：域是一组有逻辑关系的资源，可以将其作为一个单元进行管理。可以将应用程序作为域的一部分进行部署和管理。

在“运行环境”窗口中，展开“服务器”节点，右键单击 Sun Java System Application Server 实例节点。选择弹出菜单中的“查看管理控制台”命令，在 NetBeans 集成开发环境默认的 Web 浏览器中打开管理控制台的登录界面，如图 6-27 所示。

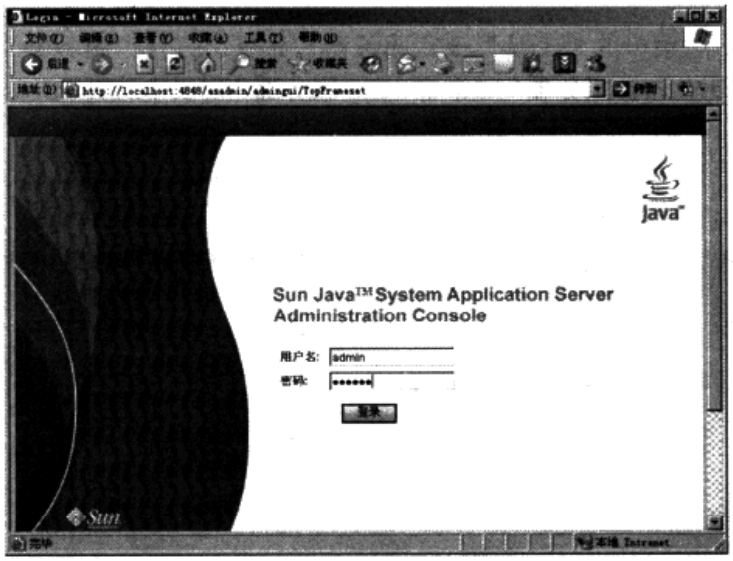


图 6-27 Sun Java Application Server 管理控制台登录界面

在输入用户名和密码后，单击“登录”按钮，就会进入管理控制台，如图 6-28 所示。在这个控制台中可以查看日常任务、管理应用程序、配置和部署 Web 服务、部署和注册自定义 MBean、管理各种资源、配置服务器等。

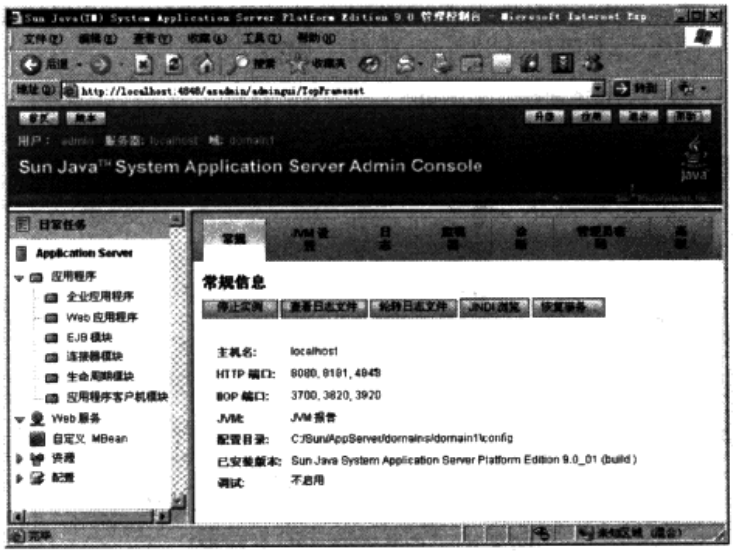


图 6-28 Sun Java Application Server 管理控制台

Download at http://www.pin5i.com/

6.3 JBoss 应用程序服务器

JBoss Application Server 4 支持基于 JavaServer Pages（JSP）、Java Servlets 和 Enterprise JavaBeans（EJB）组件的应用程序。可以使用此服务器来执行以下操作：

- 部署企业应用程序和 Enterprise Bean；
- 部署 Web 服务和使用 JSR-101 模块的 Web 服务客户端；
- 部署 Web 应用程序、JSP 页和 Servlet。

6.3.1 JBoss 服务器的注册

选择 NetBeans 主菜单的“工具”→“服务器”命令，打开“服务器”对话框。单击“添加服务器”按钮，打开“选择服务器”对话框。在“服务器”列表中选择 JBoss Application Server，如图 6-29 所示。

单击“下一步”按钮，打开“服务器位置”对话框，如图 6-30 所示。在“安装位置”文本框中输入 JBoss Application Server 的本地安装路径，也可以单击“浏览”按钮选择该位置。

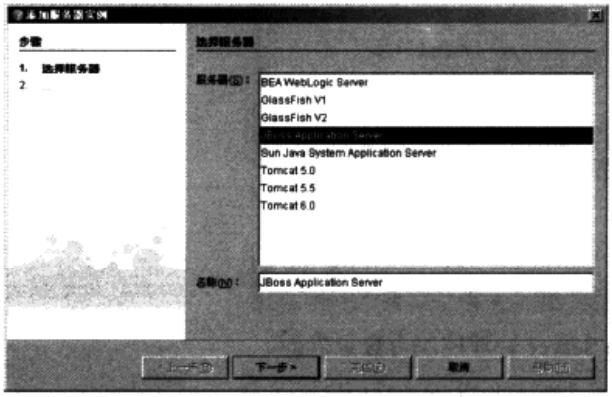


图 6-29 “选择服务器”对话框

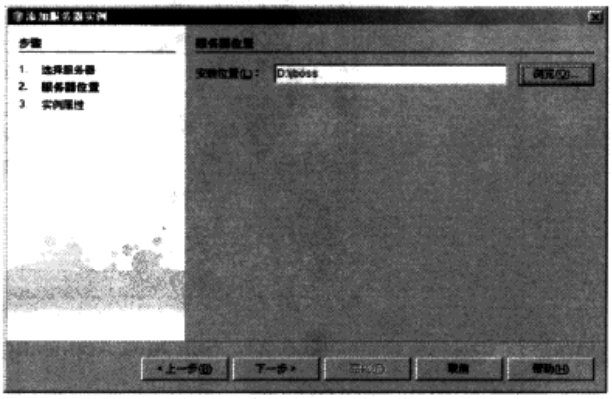


图 6-30 “服务器位置”对话框

单击“下一步”按钮，打开“实例属性”对话框，如图 6-31 所示。该对话框可以用来指定要注册的域。从“域”下拉列表框中选择该域。在“域路径”文本框中显示了 JBoss 服务器域使用的本地路径，该属性不可以在这个对话框中修改。

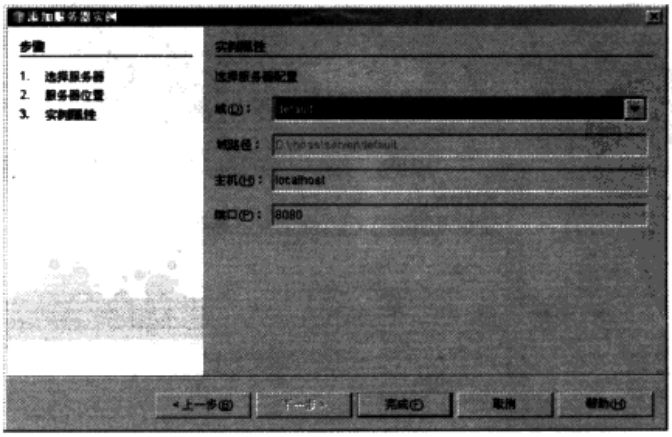


图 6-31 “实例属性”对话框

“主机”文本框中显示主机的名字，该属性不可以在这个对话框中进行修改。“端口”文本

Download at http://www.pin5i.com/

框中显示的是 JBoss 服务器使用的端口号，该属性也不可以在这个对话框中进行修改。



说明：如果要更改某个域的端口号，则必须在该域的 server.xml 文件中完成此操作。默认情况下，端口号为 8080。例如，如果使用 default 域，则可以在 JBoss 安装目录的 \server\default\deploy\jbossweb-tomcat55.sar\文件夹中找到该域的 server.xml 文件进行修改。



6.3.2 JBoss 服务器的管理控制台

在“运行环境”窗口中，展开“服务器”节点，右键单击 JBoss Application Server 4 实例节点。选择弹出菜单的“查看管理控制台”命令，在 NetBeans 集成开发环境默认的 Web 浏览器中打开管理控制台，如图 6-32 所示。

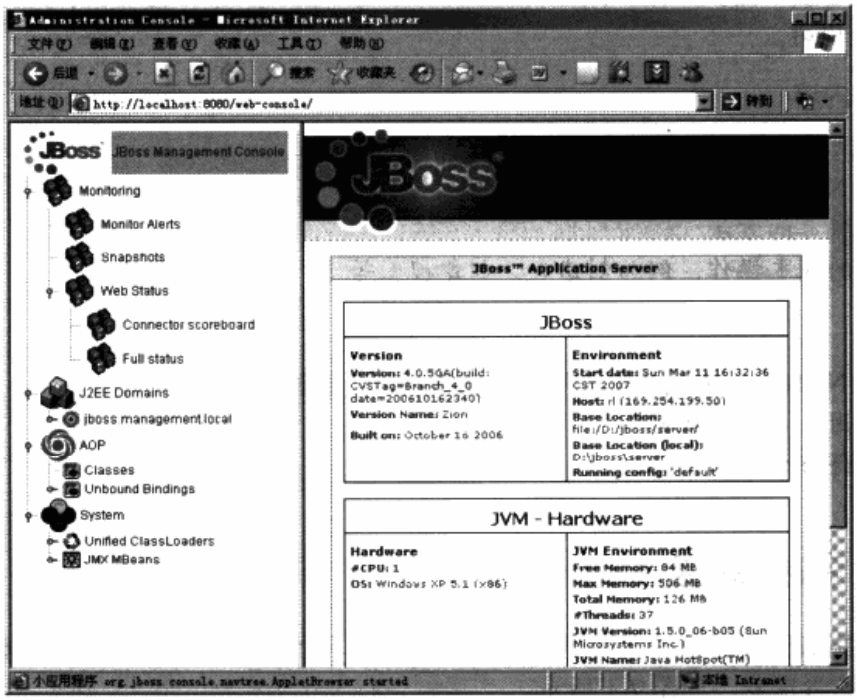


图 6-32 JBoss 服务器的管理控制台



6.4 小结

本章在全书中具有承上起下的作用，本章中安装的服务器会在后续各章中被使用。例如开发 Web 程序时会使用到 Tomcat 服务器，开发 Java EE 程序时会使用到 Sun Java 应用程序服务器或者 JBoss 服务器。

NetBeans 服务器具有较好的支持，几个主流的服务器都可以集成到 NetBeans 集成开发环境中。本章分别给出了 Tomcat、Sun Java 应用程序服务器和 JBoss 服务器的注册和配置管理方法。在实际开发中，读者可以根据实际的服务器，选择相关的小节进行学习。

Download at http://www.pin5i.com/

第7章

数据库的管理与开发

NetBeans 集成开发环境提供了对数据库开发的支持，用户可以执行 SQL 语句或者可视化地对数据库进行操作。NetBeans 对目前大多数的主流数据库都提供了支持，例如 SQL Sever 数据库、Access 数据库、MySQL 数据库、Java DB 数据库等。

本章按照使用的顺序，介绍了数据库驱动程序的配置、数据库的连接、数据表的操作以及如何执行 SQL 语句。

本章主要内容

- 如何配置 JDBC 驱动程序
- 如何在 NetBeans 中注册 Java DB 数据库
- 连接数据库的方法
- 数据库的基本操作，例如创建表、创建视图等

本章亮点

- 如何在 NetBeans 集成开发环境中配置 MySQL 数据库
- 如何使用 Java DB 数据库
- 如何获得持久性数据库结构

7.1 数据库及驱动管理

NetBeans 集成开发环境支持数据库的快速开发，在“服务”窗口中展开“数据库”节点，打开数据库资源管理器，如图 7-1 所示。

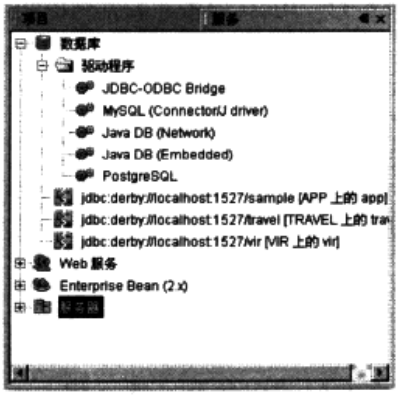


图 7-1 数据库资源管理器

在“服务”窗口中“数据库”节点的“驱动程序”节点下，会列出所有的数据库驱动程序。如果在该节点下没有发现期望的驱动程序，可以通过下面的步骤进行添加。

❶ 在“服务”窗口中展开“数据库”节点。

Download at http://www.pin5i.com/

❷ 右键单击“驱动程序”节点，然后选择弹出菜单中的“新建驱动程序”命令，打开“新建 JDBC 驱动程序”对话框，如图 7-2 所示。

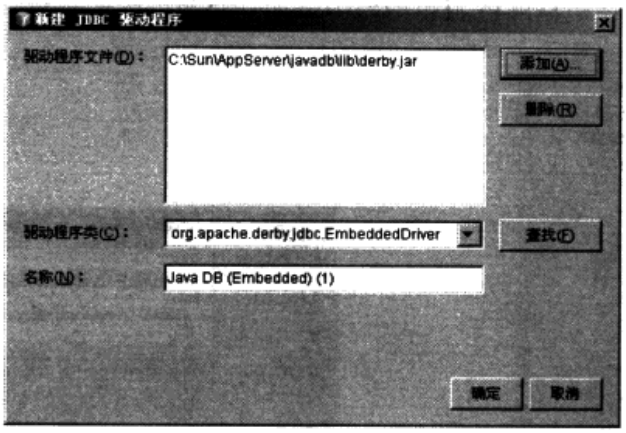


图 7-2 “新建 JDBC 驱动程序”对话框

❸ 在“新建 JDBC 驱动程序”对话框中，单击“添加”按钮以打开文件选择器，查找并选择要添加的驱动程序的归档文件。

❹ 在“驱动程序类”下拉列表框中，输入或者选择驱动程序的类。如果“驱动程序类”字段为空或不正确，可以单击“查找”按钮。集成开发环境将搜索在第❸步中选定的归档文件，并查找实现 JDBC API 驱动程序接口（java.sql.Driver）的所有类。查找成功后，会在“驱动程序类”下拉列表框中列出找到的类。

❺ 在“名称”文本框中输入新驱动程序的描述名称。集成开发环境将此名称用于显示的目的。

❻ 单击“确定”按钮，将新的驱动程序添加到注册的驱动程序列表中。注册的驱动程序的图标为 。

如果要删除某个驱动程序，可以在“服务”窗口中右键单击要输出的驱动程序节点，然后选择弹出菜单中的“删除”命令。

如果要修改驱动程序配置，可以在“服务”窗口中右键单击要修改的驱动程序节点，然后选择弹出菜单中的“定制”命令。这样会打开如图 7-2 所示的“新建 JDBC 驱动程序”对话框。在该对话框中，用户可以修改驱动程序的配置。

7.2 创建 Java DB 数据库

Java DB 数据库是 Sun 支持的开放源代码 Apache Derby 数据库版本。在 NetBeans 5.x 中，Java DB 数据库服务器与 Sun Java System Application Server 9.0 Platform Edition 捆绑在一起，替代了与 Sun Java System Application Server 以前版本捆绑在一起的 PointBase 数据库服务器。在 NetBeans 6.0 中，Java DB 数据库已经集成在了 NetBeans 集成开发环境中。

Java DB 是使用纯 Java 编程语言编写的，是完全事务性的、安全且基于标准的数据库服务器，完全支持 SQL、JDBC API 和 Java EE 技术。

NetBeans 集成开发环境主窗口中的“工具”下面包含一个“Java DB 数据库”菜单项，如图 7-3 所示。可以使用此菜单项，方便地启动和停止 Java DB 数据库服务器并创建新的数据库。

选择“工具”→“Java DB 数据库”→“创建数据库”命令，打开“创建 Java DB 数据库”对话框，可以用来创建一个新的 Java DB 数据库，如图 7-4 所示。

Download at http://www.pin5i.com/

在“数据库名称”文本框中输入数据库的名字，在“用户名”文本框中输入登录数据库使用的用户名，在“口令”文本框中输入用户的密码。单击“确定”按钮，完成数据库的创建。创建数据库后，会自动在数据库资源管理器中添加一个数据库的连接。

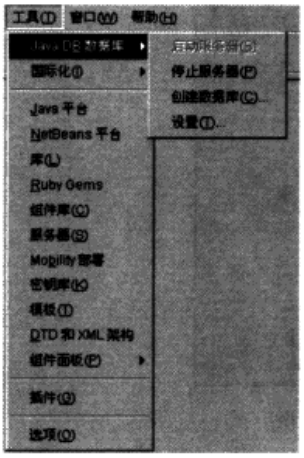


图 7-3 “Java DB 数据库”菜单项

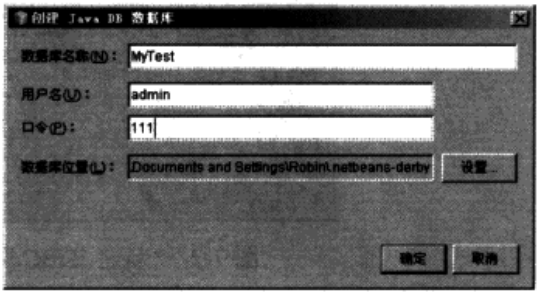


图 7-4 “创建 Java DB 数据库”对话框

7.3 连接数据库

在 NetBeans 集成开发环境中要连接数据库，首先要确定在集成开发环境中注册了数据库的驱动程序，然后才能创建数据库连接，并连接数据库。

7.3.1 连接 Java DB 数据库

在对 Java DB 数据库进行操作时，要先和数据库建立连接。下面是连接 Java DB 数据库的具体操作步骤。

❶ 选择“工具”→“Java DB 数据库”→“启动服务器”命令，启动 Java DB 数据库。在“输出”窗口中，显示如图 7-5 所示的信息，这表示数据库服务器已经启动，并且监听 1527 端口。

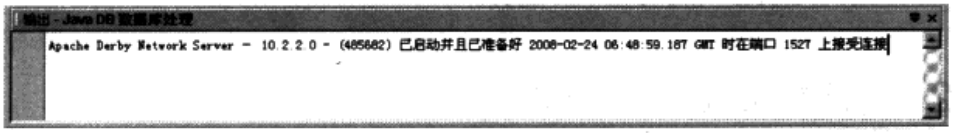


图 7-5 数据库服务器启动的信息

❷ 在“服务”窗口中，右键单击“数据库”节点，然后选择弹出菜单中的“新建连接”命令，打开“新建数据库连接”对话框，如图 7-6 所示。

❸ 在“名称”下拉列表框中可以选择数据库连接的名称，这里有两个选项，其含义如下所示。

- Java DB（Network）：需要从多个 JVM 连接数据库时，请使用此驱动程序。通过使用网络驱动程序，还可以从远程计算机访问数据库。
- Java DB（Embedded）：如果确定应用程序将从单个 JVM 中访问数据库，请使用此驱动程序。这种情况的一个典型示例是，使用 Java DB 数据库作为数据存储的单用户 Swing JDBC 应用程序。

Download at http://www.pin5i.com/

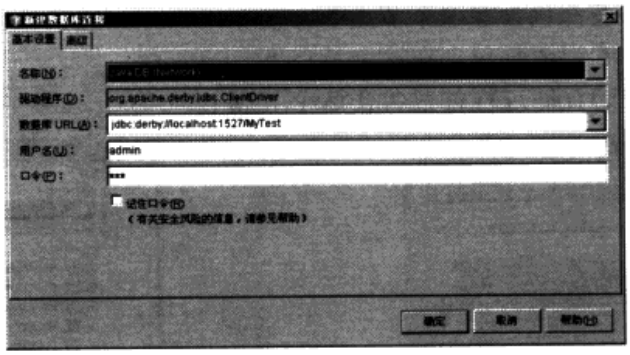


图 7-6 “新建数据库连接”对话框

❹ 在“数据库 URL”文本框中输入数据库的连接字符串，其格式定义如下面的第一行代码。其中“<主机>”必须给出，它是数据库所在的计算机名或者 IP 地址，对于本机可以输入 localhost 或者 127.0.0.1。“:<端口>”可选，是数据库使用的端口号，由图 7-6 所示的界面中可以得到数据库服务器使用的端口。“<DB 名字>”是数据库的名字，必须给出。如果要连接的数据库位于默认 Java DB 数据库文件夹中，则只需指定数据库名称。如果数据库不在默认的数据库文件夹中，则必须输入数据库的完整路径，而不仅仅是数据库名称。如果连接数据库时需要一些特定的参数，则需要在后面的“[;属性 1=属性值 1[;...]]”给出。下面代码的第二行是一个示例的数据库 URL，表示连接本机端口 1527 的 sample 数据库。该数据库位于 home 目录的.netbeans-derby 目录中。

```
jdbc:derby://<主机>[:<端口>]/<DB 名字>[;属性 1=属性值 1[;...]]
jdbc:derby://localhost:1527/sample
```

❺ 在“用户名”文本框中输入登录数据库服务器的用户名，在“口令”文本框中输入用户的口令。Java DB 样例数据库的默认用户名和口令均为 app。如果选择“记住口令”复选框，则会存储当前会话的口令。如果取消选择，那么在每次连接数据库时，都会出现一个对话框，提示输入口令。

❻ 单击“确定”按钮，完成数据库连接设置。此时会在“服务”窗口添加一个连接，如图 7-7 所示。

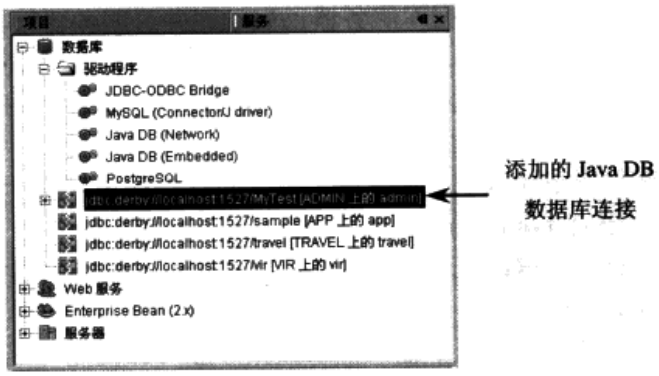


图 7-7 添加的 Java DB 数据库连接

❼ 在“服务”窗口中，右键单击添加的 Java DB 数据库连接节点，选择弹出菜单中的“连接”命令。如果在创建连接时没有选中“记住口令”复选框，那么会显示如图 7-8 所示的“连接”对话框。

❽ 输入正确的用户名和口令后，单击“确定”按钮，此时会连接到指定的 Java DB 数据库，并在数据库连接节点下面添加了如图 7-9 所示的三个节点。

Download at http://www.pin5i.com/

- 表：数据库中的表都包含在这个节点下。
- 视图：数据库中的视图都包含在这个节点下。
- 过程：数据库中的存储过程都包含在这个节点下。

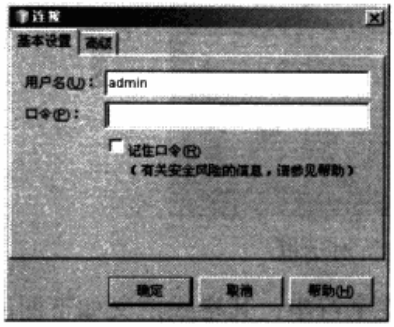


图 7-8 “连接”对话框

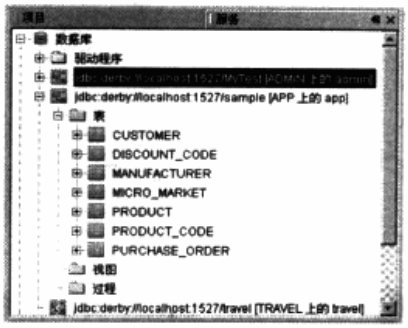


图 7-9 数据库的内容

在连接到数据库后，就可以对数据库进行操作了，例如添加数据表、修改数据、执行查询等。这些具体内容都将在后续几节中进行介绍。

7.3.2 Pointbase 数据库的注册和连接

Pointbase 数据库是与 Sun Java System Application Server 8.1 捆绑在一起的样例数据库。安装 Sun Java System Application Server 8.1 时，将自动安装 Pointbase 数据库 JDBC 驱动程序，并将 Pointbase 数据库服务器安装到 Sun/AppServer/pointbase 目录中。目前已经有版本比较新的数据库以及 Sun Java System Application Server，但是考虑到很多用户仍然使用这个数据库，因此本节将对这个数据库的注册以及连接进行简单的介绍。

在 NetBeans 集成开发环境中注册 Sun Java System Application Server 8.1 的实例时，将自动在“服务”窗口中注册样例数据库，并在“工具”菜单中显示“PointBase 数据库”菜单。可以使用这个菜单中的命令来启动和停止 PointBase 数据库。连接到 Pointbase 样例数据库的具体操作步骤介绍如下。

❶ 选择“工具”→“Pointbase 数据库”→“启动本地 Pointbase 数据库”命令，启动数据库服务器。确认 Pointbase 数据库服务器已启动，并且服务器正在侦听某个端口。

❷ 在“服务”窗口中，展开“数据库”节点和“驱动程序”节点。

❸ 右键单击 Pointbase 驱动程序节点，然后选择弹出菜单中的“连接方法”命令，打开如图 7-6 所示的“新建数据库连接”对话框。

❹ 在“名称”下拉列表框中选择“Pointbase”作为名称，在“驱动程序”文本框中包含 Pointbase 的驱动程序 com.pointbase.jdbc.jdbcUniversalDriver。

❺ 在“数据库 URL”文本框中输入 Pointbase 数据库的 URL，例如：jdbc:pointbase://localhost:9092/sample。

❻ 在“用户名”文本框中输入登录数据库服务器的用户名，在“口令”文本框中输入用户口令。Pointbase 样例数据库的默认用户名和口令均为 pbpublic。

❼ 单击“确定”按钮，添加 Pointbase 数据库的连接。在“服务”窗口中，右键单击添加的 Pointbase 数据库的连接节点，选择弹出菜单的“连接”命令即可连接数据库。

7.3.3 连接 MySQL 数据库

MySQL 数据库是一个非常流行的开源数据库，很多应用程序都采用了这个数据库。在

Download at http://www.pin5i.com/

NetBeans 集成开发环境使用 MySQL 数据库的方法和前面介绍的使用其他数据库的方法类似，但是鉴于 MySQL 数据库的重要性，以及有一些特殊设置，下面给出连接 MySQL 数据库的具体方法。

在 NetBeans 6.0 中已经默认添加了 MySQL 的数据库驱动程序，因此用户不必再单独添加。下面的操作步骤适用于 NetBeans 5.x 版本，或者在 NetBeans 6.0 中 MySQL 的数据库驱动程序不存在的情况（例如被误删除了）。

❶ 在安装好 MySQL 数据库后，需要到 www.mysql.com 网站上下载一个单独的 JDBC 驱动程序。下载后，把 MySQL 的 JDBC 驱动程序解压到某个目录。

❷ 在“服务”窗口中右键单击“数据库”→“驱动程序”节点，然后选择“新建驱动程序”命令，打开“新建 JDBC 驱动程序”对话框，如图 7-10 所示。

❸ 单击“添加”按钮，选择前面下载的 MySQL 数据库的 JDBC 驱动程序。单击“查找”按钮，自动添加驱动程序类以及名称。单击“确定”按钮，完成 MySQL 数据库的驱动程序设置。

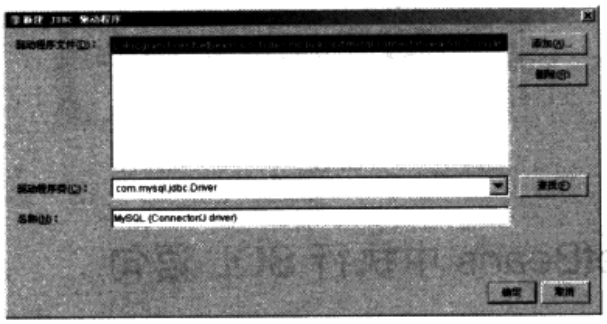
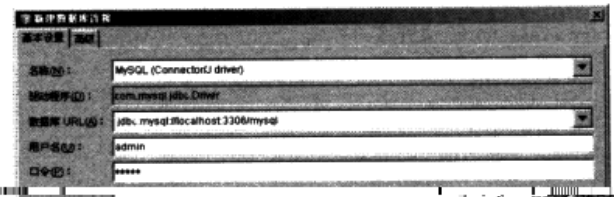


图 7-10　为 MySQL 数据库添加驱动

下面介绍连接 MySQL 数据库的方法。

❶ 启动 MySQL 数据库。为 MySQL 数据库添加一个用户（非 ROOT 用户）。

❷ 在“服务”窗口中，右键单击“数据库”节点，然后选择弹出菜单的“新建连接”命令，打开“新建数据库连接”对话框，如图 7-11 所示。



Download at http://www.pin5i.com/

❹ 在“用户名”文本框中输入在第 2 步创建的 MySQL 数据库的用户名，在“口令”文本框中输入用户口令。如果选择“记住口令”复选框，则会存储当前会话的口令。如果取消选择，那么在每次连接数据库时，都会出现一个对话框，提示输入口令。

❺ 单击“确定”按钮，完成数据库连接设置。在“服务”窗口中，右键单击添加的 MySQL 数据库连接节点，选择弹出菜单中的“连接”命令。连接到 MySQL 数据库后，会显示如图 7-12 所示的界面。

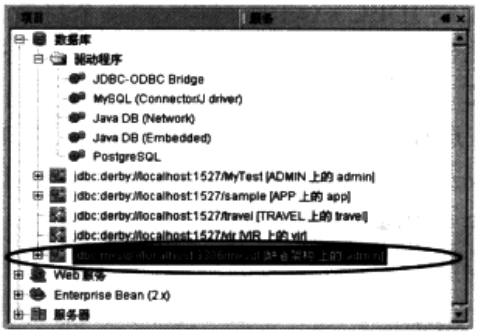


图 7-12　连接到 MySQL 数据库

7.4　在 NetBeans 中执行 SQL 语句

NetBeans 集成开发环境中支持直接执行 SQL 语句，实现数据库的查询或者操纵。在“服务”窗口中展开“数据库”节点。右键单击连接的数据库节点，然后从弹出菜单中选择“执行命令”，可以打开 SQL 编辑器，如图 7-13 所示。

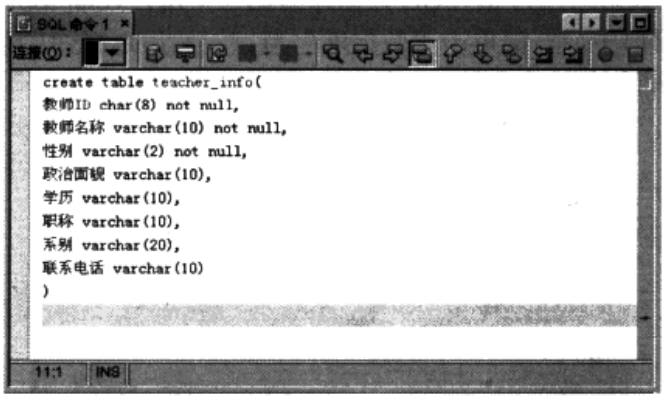


图 7-13　SQL 编辑器

说明：在“服务”窗口中，可以从连接的数据库的任何子节点中执行“执行命令”弹出式菜单命令，打开 SQL 编辑器。

在“连接”下拉列表框中可以选择不同的数据库连接，SQL 语句只对当前选中的数据库有效。在下面的文本编辑区域可以输入要执行的 SQL 语句，例如输入创建数据表 teacher_info 的语句如下所示：

```
create table teacher_info(
教师 ID char(8) not null,
教师名称 varchar(10) not null,
性别 varchar(2) not null,
```

Download at http://www.pin5i.com/


```
政治面貌 varchar(10),
学历 varchar(10),
职称 varchar(10),
系别 varchar(20),
联系电话 varchar(10)
)
```

输入完毕后，单击 SQL 编辑器工具栏的“运行 SQL”按钮，执行 SQL 语句。如果语句执行成功会在“输入”窗口中显示如图 7-14 所示的提示信息。

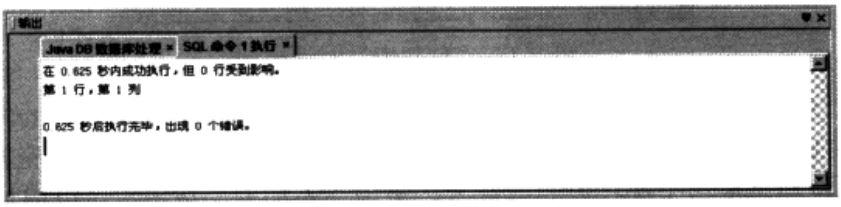
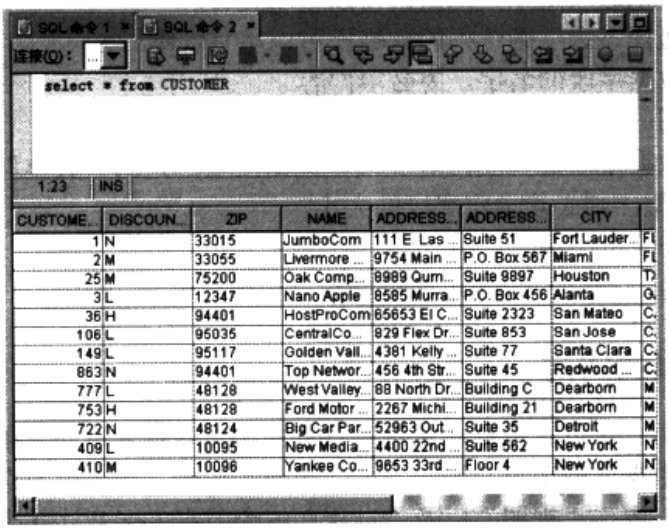


图 7-14 在“输入”窗口中 SQL 语句执行成功的提示信息

对于有输出结果的 SQL 语句，会在 SQL 编辑器的下半部分显示执行的结果。例如执行 SQL 查询语句 select * from CUSTOMER，返回 CUSTOMER 数据表中的全部内容，那么执行结果如图 7-15 所示。



CUSTOME...	DISCOUN...	ZIP	NAME	ADDRESS...	ADDRESS...	CITY	
1	N	33015	JumboCom	111 E. Las ...	Suite 51	Fort Lauder...	Fl
2	M	33055	Livermore ...	9754 Main ...	P.O. Box 567	Miami	Fl
25	M	75200	Oak Comp...	8989 Gurn...	Suite 9897	Houston	Tx
3	L	12347	Nano Apple	8585 Murra...	P.O. Box 456	Alanta	Ga
36	H	94401	HostProCom	65653 El C...	Suite 2323	San Mateo	Ca
106	L	95035	CentralCo...	929 Flex Dr...	Suite 853	San Jose	Ca
149	L	95117	Golden Vall...	4381 Kelly ...	Suite 77	Santa Clara	Ca
863	N	94401	Top Networ...	456 4th Str...	Suite 45	Redwood ...	Ca
777	L	48128	West Valley...	88 North Dr...	Building C	Dearborn	MI
753	H	48128	Ford Motor ...	2267 Michi...	Building 21	Dearborn	MI
722	N	48124	Big Car Par...	52963 Out...	Suite 35	Detroit	MI
409	L	10095	New Media...	4400 22nd ...	Suite 562	New York	NY
410	M	10096	Yankee Co...	9653 33rd ...	Floor 4	New York	NY

图 7-15 显示执行结果

7.5 数据表的基本操作

除了可以使用 SQL 语句完成数据库的操作之外，NetBeans 集成开发环境还提供了一些图形界面以完成一些操作，例如创建数据表、添加/删除列、查询表中的数据等。本节将具体介绍这些基本的操作步骤。

7.5.1 创建数据表

在连接到数据库后，可以在数据库资源管理器中创建表。在“服务”窗口中，展开已连接的数据库的节点。右键单击“表”节点，然后选择弹出菜单中的“创建表”命令，打开“创建表”对话框，如图 7-16 所示。

Download at http://www.pin5i.com/

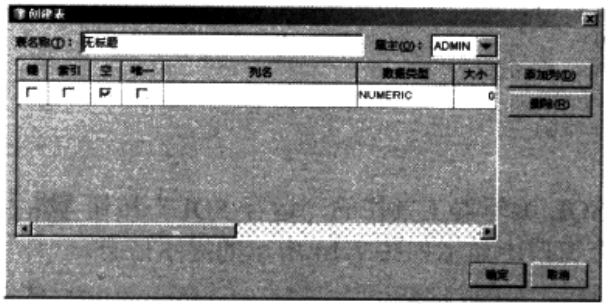


图 7-16 “创建表”对话框

在“表名”文本框中输入表的名称。此名称必须符合数据库中引用表名的约定。通常区分大小写，并且可以使用空格。但是，建议以大写形式输入表名，并且不包含空格。

“属主”下拉列表框用于选择表的所有者。单击“添加列”按钮，可以增加新的一列。单击“删除列”按钮可以删除一列。对于某一列，可以在左下角的网格中进行修改和编辑。其中各个列的含义如下所示。

- “密钥”列：选择该选项则会为列建立密钥。
- “索引”列：选择该选项则会为列建立索引。
- “空”列：选择该选项则该列的数据可以为空。
- “唯一”列：选择该选项则该列的数据不可以重复（唯一）。
- “列名”列：数据字段的名字，为必填内容。
- “数据类型”列：数据字段的数据类型，为必填内容。
- “大小”列：数据字段的长度，对于某些数据类型，列的“大小”选项是可选的，而对于一些数据类型，则必须指定字段的大小。
- “比例”列：列所占比例。

输入完毕后，单击“确定”按钮，即可创建一个数据表。

7.5.2 浏览表的结构

在连接到数据库后，可以在数据库资源管理器中浏览数据表以及其他对象。在“服务”窗口中，展开已连接的数据库的节点。一个数据库一般是由表、视图以及存储过程组成，因此在数据库节点下都包含“表”、“视图”以及“过程”三个节点。展开这几个节点可以查看更进一步的信息。

展开“表”节点，可以看到数据库中的所有表。再展开某个表，还可以看到表中的字段、索引以及外键等信息，如图 7-17 所示。

单击某个索引节点可查看该索引包含的列的列表。单击某个外键节点可查看该外键包含的列的列表。外键包含的列引用另一个表的列（通常为主键）。

展开“视图”节点，可以查看数据库中的所有视图。视图列以灰色显示，表示不能更改视图列的物理属性。但是，可以通过先删除视图再重新创建它来更改视图。展开“过程”节点，可以查看存储的过程名称及其 in 和 out 参数。

右键单击某个对象，选择弹出菜单中的“属性”命令，可以打开“属性”对话框，查看对象的属性，如图 7-18 所示。

7.5.3 添加/删除列

在数据库资源管理器中，还可以对已创建的表的结构进行调整，例如添加/删除列。在“服

Download at http://www.pin5i.com/

务”窗口中，展开已连接的数据库的节点，选择要删除的数据表的列。然后右键单击该列，选择弹出菜单中的“删除”命令，即可删除指定的列。



图 7-17 在数据库资源管理器中查看数据库中的表

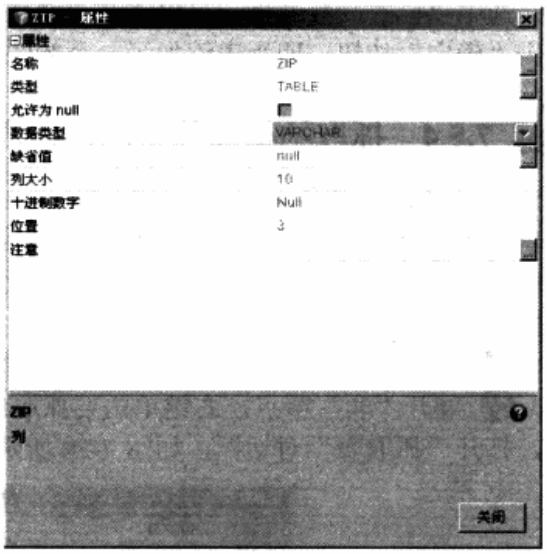


图 7-18 查看对象的属性

说明：实际上，在数据库资源管理器中通过上下文弹出菜单，也可以删除其他的数据库对象，例如数据表、视图、外键等。

如果要添加某一个列，可以在要添加列的数据表上单击右键，选择弹出菜单中的“添加列”命令，打开“添加列”对话框，如图 7-19 所示。

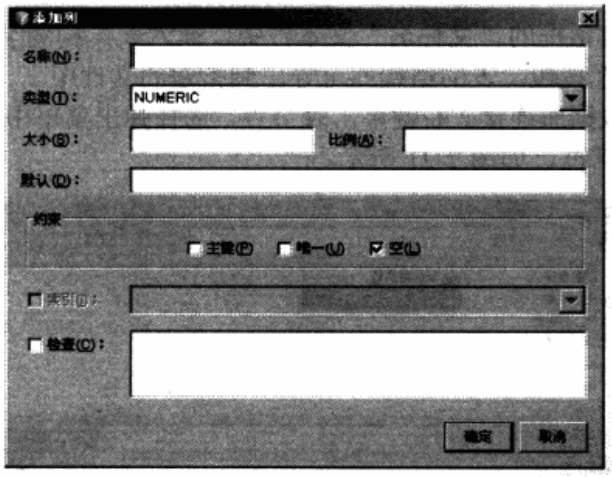


图 7-19 “添加列”对话框

在“名称”文本框中输入列（字段）的名称。此名称必须符合数据库中引用的字段名的约定。在“类型”下拉列表框中可以选择列的数据类型。某些数据类型需要在“大小”文本框中输入数据类型的大小。

在“默认”文本框中可以给出字段的默认值。“约束”分组框中可以设置列的约束，具体含义如下所示。

- “主键”复选框：设置字段为主键。单击该复选框时，也会同时选中“唯一”复选框以及“索引”复选框。

Download at http://www.pin5i.com/

- “唯一”复选框：该字段的数据不可以重复（唯一）。
- “空”复选框：选择该选项则该字段的数据可以为空。

选中“检查”复选框，可以在旁边的文本框中输入该字段的其他约束条件。输入完毕后，单击“确定”按钮，即为表添加了一个新的列。

7.5.4 抓取结构以及重新创建表

在 NetBeans 的数据库资源管理器中，可以把表的结构进行备份（导出），以便于将来可以重新快速地创建表（导入），或者把这个表移动到其他的数据库中。下面以一个操作实例来说明如何把一个数据库中的表迁移到另一个数据库中。注意，此方法只能复制数据表的结构，不能复制数据。

❶ 在“服务”窗口中，展开已连接的数据库的节点。

❷ 展开“表”节点，右键单击要抓取结构的表，然后选择弹出菜单中的“抓取结构”命令，打开“抓取表”对话框，如图 7-20 所示。

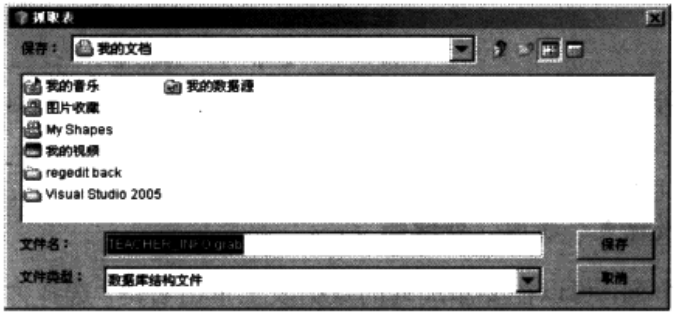


图 7-20 “抓取表”对话框

❸ 在“抓取表”对话框的“文件名”文本框中，输入表结构的名称并找到要保存文件的位置，然后单击“保存”按钮，表的结构即可导出到指定的文件中。

❹ 打开另外一个数据库连接。右键单击表节点，选择弹出菜单中的“重新创建表”命令，打开“重新创建表”对话框，如图 7-21 所示。

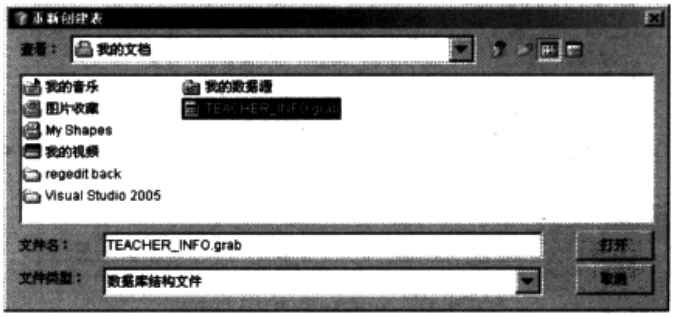


图 7-21 “重新创建表”对话框

❺ 在“重新创建表”对话框中，选择数据库结构文件，然后单击“打开”命令，打开“命名表”对话框，如图 7-22 所示。

❻ 在“名称”文本框中输入新表的名字。这里可以选择与原数据表不同的名字。在“创建表脚本”文本框中，显示了要执行的操作对应的 SQL 语句。如果需要修改 SQL 语句，可以单击“编辑脚本”按钮，进入编辑模式。单击“确定”按钮，完成数据表的重新创建。

Download at http://www.pin5i.com/

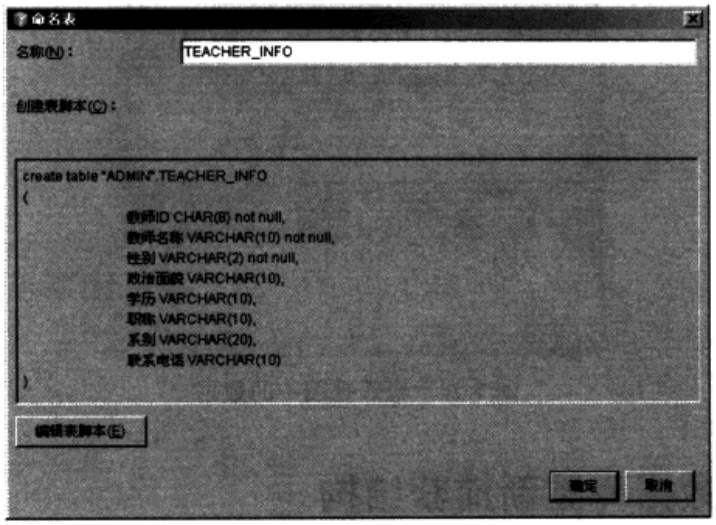


图 7-22 “命名表”对话框

7.6 视图操作

在连接到数据库后，可以在数据库资源管理器中创建视图。在“服务”窗口中，展开已连接的数据库的节点。右键单击“视图”节点，然后选择弹出菜单中的“创建视图”命令，打开“创建视图”对话框，如图 7-23 所示。

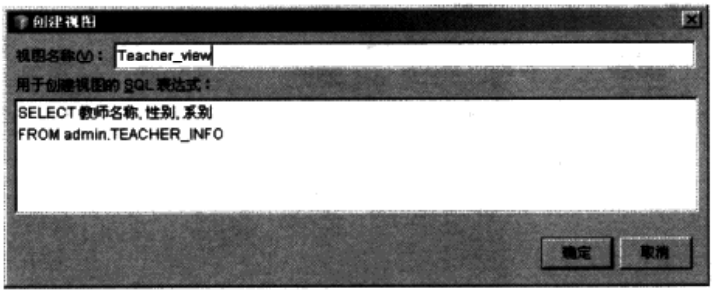


图 7-23 “创建视图”对话框

在“视图名称”文本框中输入新建视图的名字，然后在下面的“用于创建视图的 SQL 表达式”文本框中输入 SQL 表达式。注意，这里不需要给出创建视图的全部 SQL 表达式，只需要给出 SELECT 部分即可。例如，要创建一个视图的全部表达式为：

```
CREATE VIEW Teacher_view (教师名称,性别,系别)
AS
SELECT 教师名称, 性别, 系别
FROM TEACHER_INFO
```

这里，只需要给出下面的部分即可。

```
SELECT 教师名称, 性别, 系别
FROM TEACHER_INFO
```

单击“确定”按钮，即可创建一个视图。如果要查看视图中的数据，则可以在数据库资源管理器中右键单击创建的视图，选择弹出菜单中的“查看数据”命令。NetBeans 会自动创建查询 SQL 语句并执行，执行结果如图 7-24 所示。

Download at http://www.pin5i.com/

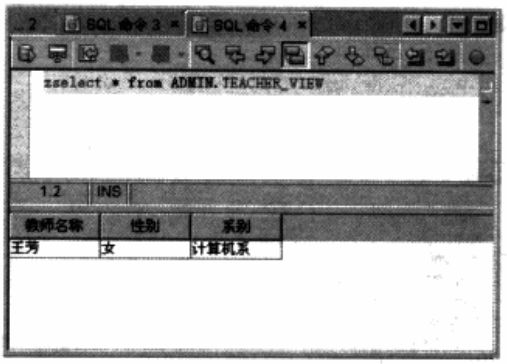


图 7-24 查看视图中的数据

7.7 从数据库重新捕获结构

NetBeans 集成开发环境允许捕获和创建持久性数据库结构，这样就可以在其他项目中使用该结构了。在数据库服务器运行的情况下可以执行下面的操作，以捕获数据库结构。

❶ 新建一个 Java 标准项目，并设置为主项目。

❷ 选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”列表中选择“持久性”节点，然后选择右边“文件类型”列表中的“数据库架构”节点，创建持久性数据库结构文件。单击“下一步”按钮，打开“名称和位置”对话框，如图 7-25 所示。

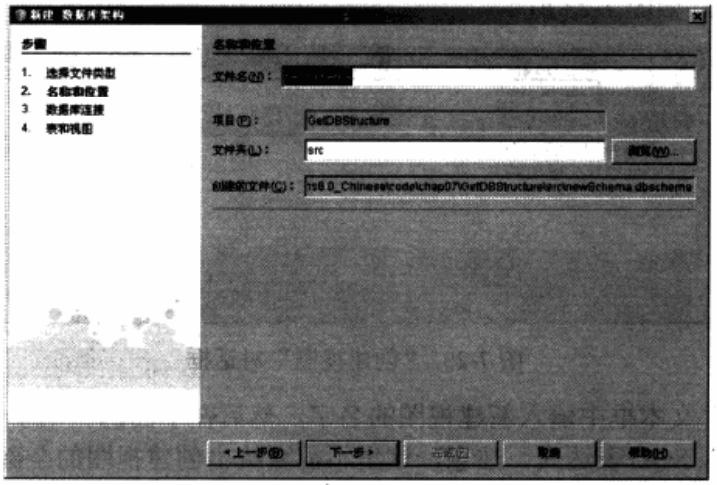


图 7-25 “名称和位置”对话框

❸ 在“文件名”文本框中输入新的持久性数据库结构文件名。如果希望把数据库结构文件创建到其他的非默认目录，可以在“文件夹”文本框中输入或者单击“浏览”按钮进行选择。单击“下一步”按钮，打开“数据库连接”对话框，如图 7-26 所示。

❹ 在下拉列表框中选择要使用的数据库连接，然后单击“下一步”按钮，如果数据库没有选择自动保存密码，那么会打开“连接”对话框，如图 7-27 所示。如果自动保存密码，则直接进入第❼步。

❺ 输入正确的用户名以及口令。在连接数据库时，与用户名同名的结构将被设置为默认结构。但如果不存在同名结构或者希望使用的其他结构，可以打开“高级”标签，并单击“获取架构”按钮。此时会自动读取数据库中所有的结构，并显示在“选择架构”下拉列表框中，如图 7-28 所示。

Download at http://www.pin5i.com/

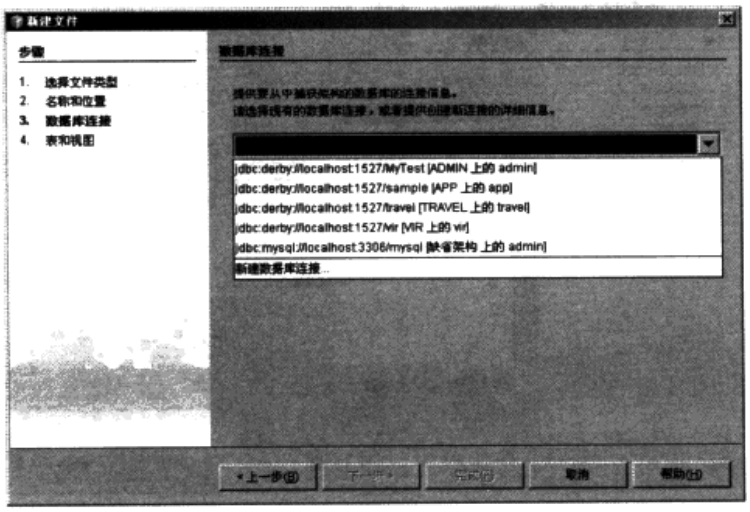


图 7-26 “数据库连接”对话框

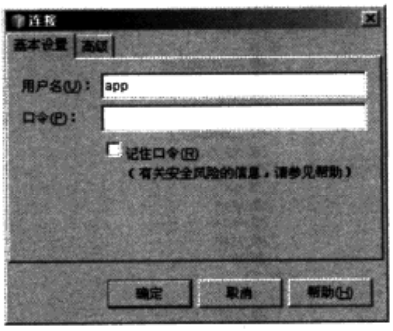


图 7-27 “连接”对话框

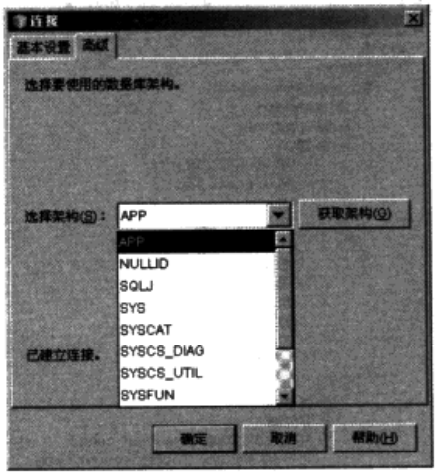


图 7-28 选择架构

❻ 在这里，可以选择一个结构，然后单击“确定”打开“表和视图”对话框，如图 7-29 所示。

❼ 在“可用的表和视图”列表中，列出了所有可以获取结构的表和视图。单击某个视图，再单击“添加”按钮可以把选中的文件添加到“选定的表和视图”列表中。可以按下 Ctrl 键，再用鼠标单击表或者视图，选择多个表或者视图。也可以先选择某个表/视图，然后按下 Shift 单击另外一个表/视图，这样可以选择两个之间的所有表/视图。单击“全部添加”按钮可以把所有可用的表/视图添加到右边的“选定的表和视图”列表中。“删除”按钮与“添加”按钮执行相反的操作；“全部删除”按钮则与“全部添加”按钮执行相反的操作。选择完表/视图后，单击“完成”按钮，创建数据库结构文件。

此时，可以在“项目”视图或者“文件”视图中查看创建的数据库结构文件，分别如图 7-30 和图 7-31 所示。

说明：如果数据库结构发生了变化，可以在“项目”视图或者“文件”视图中的数据库结构文件上单击右键，选择弹出菜单中的“从数据库重新捕获结构”命令，刷新数据库结构文件的内容。

Download at http://www.pin5i.com/

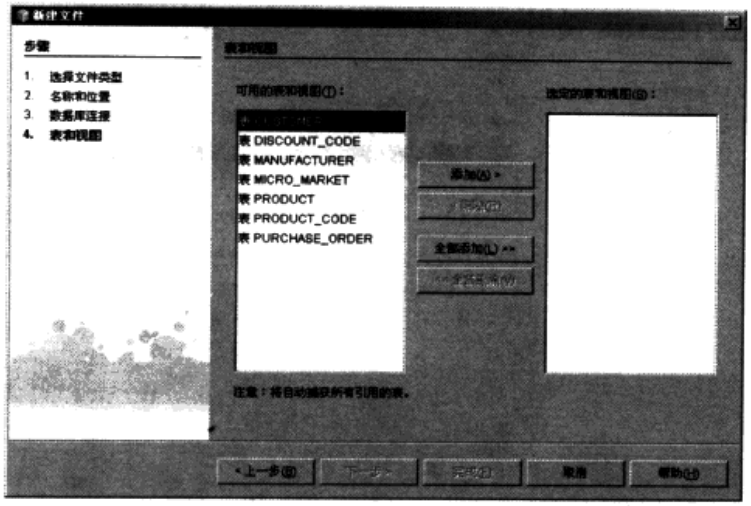


图 7-29 “表和视图”对话框

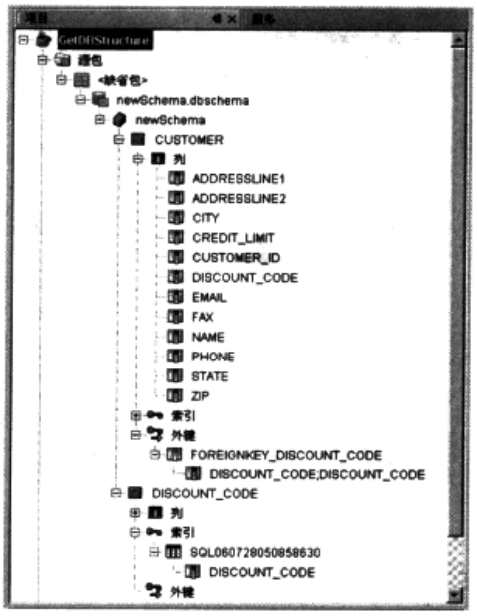


图 7-30 在“项目”视图中查看数据库结构文件

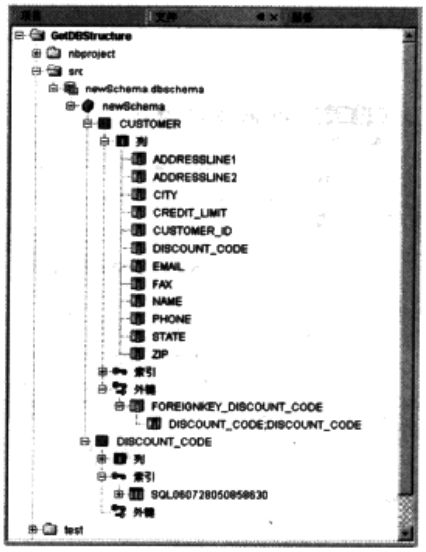


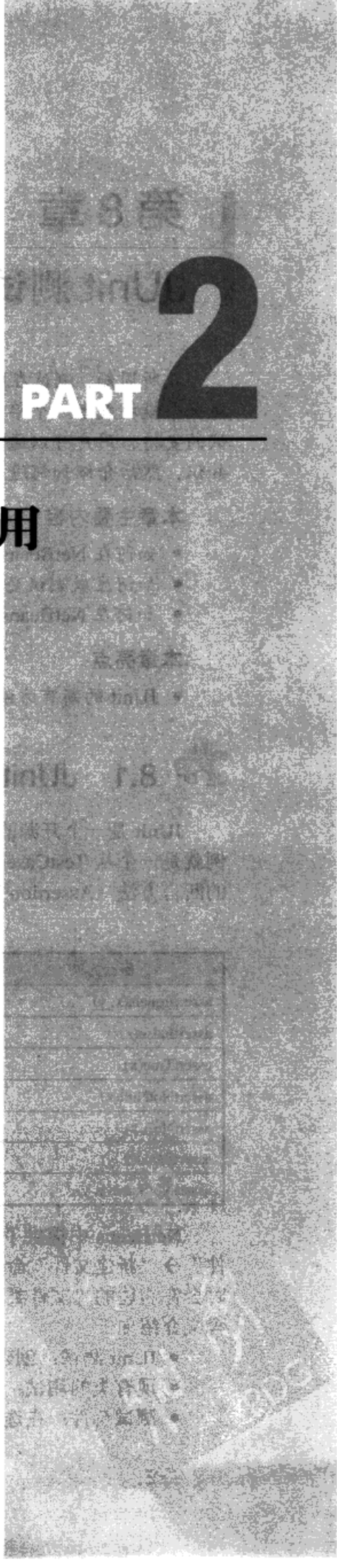
图 7-31 在“文件”视图中查看数据库结构文件

7.8 小结

本章主要介绍如何在 NetBeans 集成开发环境中对数据库进行操作。这部分内容对于开发数据库应用程序非常重要。通过本章的学习，读者应该掌握以下基本内容：

- 如何配置并连接 Java DB 数据库
- 如何配置并连接 MySQL 数据库
- 如何在 NetBeans 集成开发环境中编辑并执行 SQL 语句
- 表的创建以及如何添加列
- 表结构的导出以及导入
- 如何创建视图

Download at http://www.pin5i.com/



PART 2

第二篇　NetBeans 高级应用

第 8 章　JUnit 测试
第 9 章　性能分析
第 10 章　在 NetBeans 中进行版本控制
第 11 章　基础 Web 应用程序开发
第 12 章　高级 Web 应用程序开发
第 13 章　在 NetBeans 中开发 Struts 程序
第 14 章　进行可视化的 Web 开发
第 15 章　企业级应用程序开发
第 16 章　基于 XML 的开发
第 17 章　UML 建模
第 18 章　Web 服务以及 SOA 程序开发

Download at http://www.pin5i.com/

第8章

JUnit 测试

众所周知，测试在整个软件生命周期中占有很重要的地位。单元测试又称为模块测试，是对一个模块或者一个单元进行的测试，因此称为单元测试。NetBeans 提供了对程序进行单元测试的支持，用户可以通过 JUnit 快速完成程序单元的测试。本章将首先介绍一些 JUnit 的基础知识，然后介绍如何进行单元测试。

本章主要内容

- 如何在 NetBeans 中生成测试实例
- 如何生成测试套件
- 如何在 NetBeans 中配置 JUnit

本章亮点

- JUnit 的断言方法



8.1 JUnit 测试概述

JUnit 是一个开源的测试框架，通过这个框架可以快速地实现测试用例。一个 JUnit 测试用例就是一个从 TestCase 派生的独立的类，在类中通常包括一些测试方法。JUnit 中提供了很多的断言方法（Assertion Method）用来测试不同的条件，这些断言方法的定义如表 8-1 所示。

表 8-1 JUnit 断言方法的定义

断言方法	功能说明
assertEquals(x, y)	测试 x 是否等于 y，x 和 y 要求是简单类型数值或者必须为实现了 equals 方法的类
assertFalse(x)	测试 x 是否为假（false），x 是一个布尔类型的值
assertTrue(x)	测试 a 是否为真（true），a 是一个布尔类型的值
assertNotNull(x)	测试 x 是否非空，x 是一个对象或者 null
assertNull(x)	测试 x 是否为空，x 是一个对象或者 null
assertSame(x, y)	测试 x 和 y 是否相同
assertNotSame(x, y)	测试 x 和 y 是否不相同

NetBeans 中提供了对 JUnit 的内置支持，不用像 Eclipse 一样需要首先安装插件。选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”列表中选择 JUnit 节点，此时会在右边的“文件类型”列表中显示可以创建的 JUnit 类型，如图 8-1 所示。这几个类型的含义介绍如下。

- JUnit 测试：创建一个空的 JUnit 测试实例。
- 现有类的测试：创建一个简单的 JUnit 测试实例来测试单个类的方法。
- 测试套件：在选定的 Java 测试包中创建所有测试的测试套件。

Download at http://www.pin5i.com/

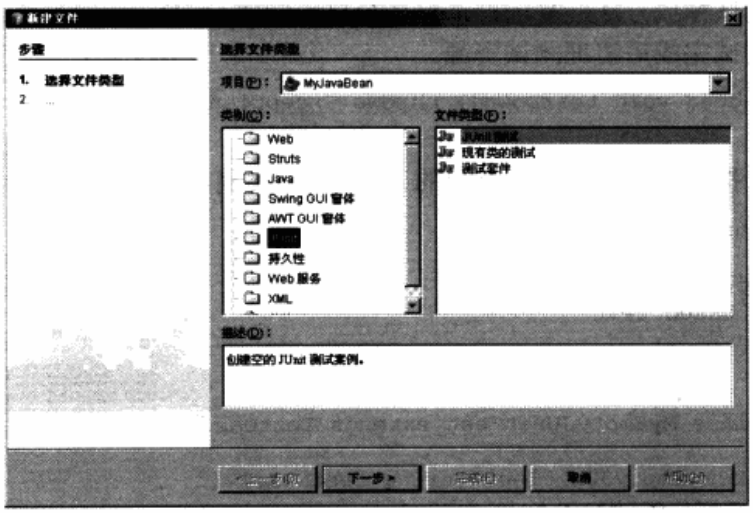


图 8-1 可以创建的 JUnit 类型

8.2 创建测试实例

创建测试实例有两种方法：一种是直接创建一个空白的测试实例，然后再添加测试代码；另一种是根据现有的类创建。下面分别对这两种方法进行详细的介绍。

8.2.1 创建空测试

在如图 8-1 所示的“新建文件”对话框中选择“JUnit 测试”节点，可以创建一个空白的测试实例。单击“下一步”按钮，打开“名称和位置”对话框，如图 8-2 所示。

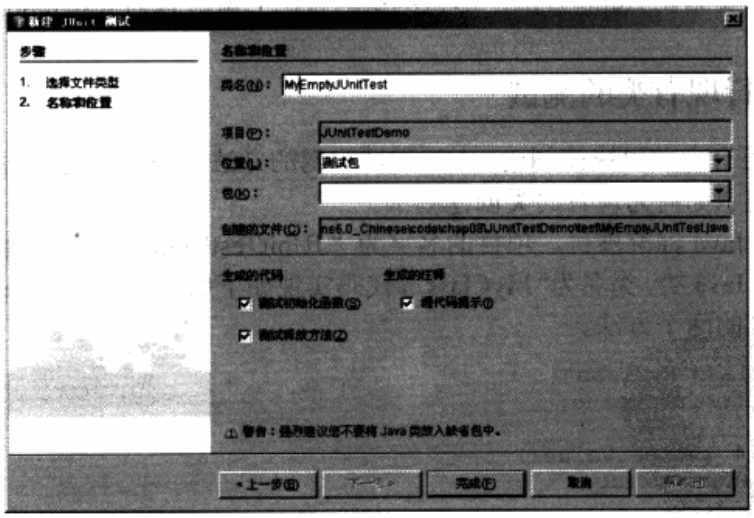


图 8-2 “名称和位置”对话框

在“类名”文本框中输入 JUnit 测试实例的类名。在“位置”下拉列表框中选择“测试包”。如果选中“测试初始化函数”复选框，那么会自动创建 setUp 空方法。在 setUp 方法中添加代码，可以用来初始化测试环境。

```
protected void setUp() throws Exception {
}
```

Download at http://www.pin5i.com/

如果选中“测试释放方法”复选框，那么会自动创建 tearDown 空方法。在 tearDown 方法中添加代码，在测试完成后清理测试资源。

```
protected void tearDown() throws Exception {
}
```

单击“完成”按钮，会创建一个空白的 JUnit 测试实例，代码如下所示。

```
import junit.framework.*;
/*
 * MyEmptyJUnitTest.java
 * JUnit based test
 *
 */

public class MyEmptyJUnitTest extends TestCase {

    public MyEmptyJUnitTest(String testName) {
        super(testName);
    }

    protected void setUp() throws Exception {
    }

    protected void tearDown() throws Exception {
    }

    // TODO 在此处添加测试方法。名称必须以 "test" 开头。例如:
    // public void testHello() {}
}
```

与 JUnit 相关的类都包含在 junit.framework 包中，因此要首先对该包进行引用。一个 JUnit 测试实例就是一个从 TestCase 派生的独立的类，在类中通常包括 setUp 方法、tearDown 方法和 testXXXX 方法。其中，testXXXX 方法就是用来进行测试的方法，XXXX 对应的是要被测试的模块的名字，例如要测试 Hello 方法，那么测试的方法名就应该为 testHello。



8.2.2 进行现有类的测试

NetBeans 中还有一种更快捷创建 JUnit 测试实例的方法，即根据现有的类创建测试实例。下面以一个实例演示如何为现有的类创建测试。

❶ 新建一个 Java 标准项目，项目的名字为“JUnitTestDemo”。将该项目设置为主项目。

❷ 创建一个 Java 类，类名为“MyClass”，代码实现如下所示。其中 getValue 方法和 setValue 方法就是要被测试的两个方法。

```
public class MyClass {
    private int value;

    public int getValue() {
        return value;
    }

    public void setValue(int value) {
        this.value = value;
    }
}
```

❸ 在“项目”窗口中，右键单击要创建测试实例的 MyClass 类，然后选择弹出菜单中的“工具”→“创建 JUnit 测试”命令，或者按下快捷键 Ctrl + Shift + U，打开“创建测试”对话框，如图 8-3 所示。

Download at http://www.pin5i.com/

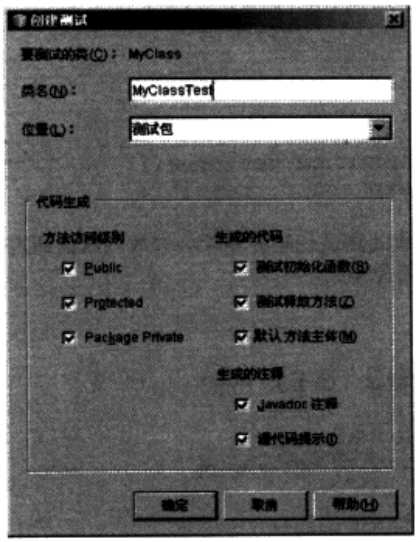


图 8-3 “创建测试”对话框

❹ 在“类名”文本框中输入测试实例的类名称。“位置”下拉列表框用于选择测试实例的保存位置，这里使用默认的“测试包”。

❺ 可以选择“Public”复选框，为公共的方法创建测试方法；选择“Protected”复选框，为受保护的方法创建测试方法；选择“Package Private”复选框，为私有的方法创建测试方法。

❻ 选中“测试初始化函数”复选框会自动创建 setUp 方法，用于初始化字段、启用日志记录以及重置环境变量等任务的方法。选中“测试释放方法”复选框会自动创建 tearDown 方法，用于在测试运行完测试后进行清理的工作。

❼ 选中“默认方法主体”复选框，将会自动创建测试方法中的代码，在此基础上再修改可以快速完成测试方法。如果未选中该复选框，那么会创建空白的测试方法。

❽ 选中“Javadoc 注释”复选框，会自动生成 Javadoc 注释；选中“源代码提示”复选框，会自动生成建议如何实现测试方法的注释。单击“确定”按钮，创建如下的测试实例。

```
import junit.framework.*;
/*
 * MyClassTest.java
 * JUnit based test
 *
 */

/**
 *
 * @author Liu Bin
 */
public class MyClassTest extends TestCase {

    public MyClassTest(String testName) {
        super(testName);
    }

    protected void setUp() throws Exception {
    }

    protected void tearDown() throws Exception {
    }

    /**
     * getValue 方法的测试（属于 MyClass 类）
     */
```

Download at http://www.pin5i.com/


```
    public void testGetValue() {
        System.out.println("getValue");

        MyClass instance = new MyClass();

        int expResult = 0;
        int result = instance.getValue();
        assertEquals(expResult, result);

        // TODO 查看生成的测试代码并删除会失败的默认调用
        fail("测试案例为原型。");
    }

    /**
     * setValue 方法的测试（属于类 MyClass）。
     */
    public void testSetValue() {
        System.out.println("setValue");

        int value = 0;
        MyClass instance = new MyClass();

        instance.setValue(value);

        // TODO 查看生成的测试代码并删除会失败的默认调用
        fail("测试案例为原型。");
    }

}
```

❾ 修改 testGetValue 方法和 testSetValue 方法中的代码如下，执行边界条件测试。

```
    public void testGetValue() {
        System.out.println("getValue");

        MyClass instance = new MyClass();

        int expResult = 0;
        int result = instance.getValue();
        assertEquals(expResult, result);
    }

    /**
     * setValue 方法的测试（属于 MyClass 类）
     */
    public void testSetValue() {
        System.out.println("setValue");

        MyClass instance = new MyClass();
        int value = Integer.MIN_VALUE;
        instance.setValue(value);
        assertEquals(instance.getValue(), value);

        value = -0;
        instance.setValue(value);
        assertEquals(instance.getValue(), value);

        value = Integer.MAX_VALUE;
        instance.setValue(value);
        assertEquals(instance.getValue(), value);
    }
```

❿ 在项目节点上单击右键，选择弹出菜单中的“测试项目”命令，开始执行单元测试。若在“输出”窗口中看到如下的输出结果，则表示测试成功了。

```
init:
```

Download at http://www.pin5i.com/


```
deps-jar:
compile:
compile-test:
getValue
setValue
Testsuite: MyClassTest
Tests run: 2, Failures: 0, Errors: 0, Time elapsed: 0.078 sec

------------- Standard Output ---------------
getValue
setValue
------------- ---------------- ---------------
test-report:
test:
生成成功（总时间：0 秒）
```



说明：也可以在“项目”或“文件”窗口中选择要测试的项目中的任何节点或文件，然后从主菜单中选择“运行”→“测试<项目名>”命令（快捷键 Alt + F6）对项目执行单元测试。NetBeans 集成开发环境将执行项目的每个测试。另外，在“项目”或“文件”窗口中选择要运行测试的类的节点。从主菜单中选择“运行”→“运行文件”→“测试<类名>”命令（快捷键 Ctrl + F6），可以只执行选中的测试实例。

⓫ 如果“JUnit 测试结果”窗口没有打开，可以在主菜单中选择“窗口”→“JUnit 测试结果”命令打开该窗口，如图 8-4 所示。在窗口中显示了测试的统计信息。如果要看输出信息，可以单击“输出”按钮。单击 按钮，则仅会显示测试失败的信息。如果测试过程中发生错误，双击错误即可跳到代码中出现该错误的行。

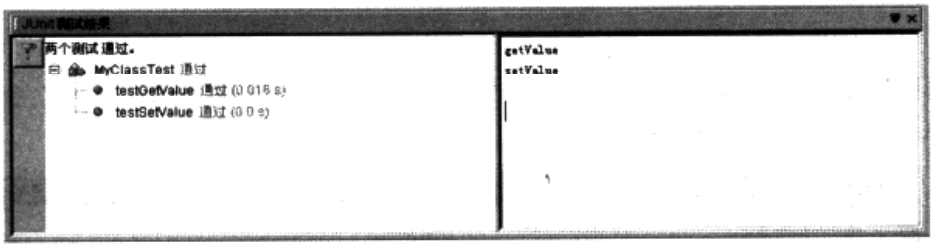


图 8-4 “JUnit 测试结果”窗口

8.3 测试套件

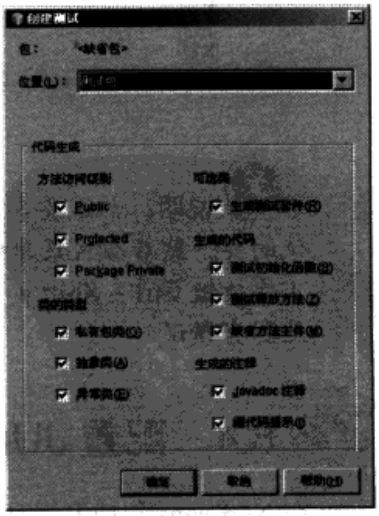


图 8-5 “创建测试”对话框

在实际开发中，往往会有很多的文件需要进行单元测试。这样可以创建 JUnit 测试套件，每个套件中可以包含一个或者多个测试实例。通过测试套件可以同时运行多个测试实例。在“项目”窗口中，右键单击包含要生成测试套件的源文件的包节点。然后选择弹出菜单中的“工具”→“创建 JUnit 测试”命令（快捷键 Ctrl + Shift + U），打开“创建测试”对话框，如图 8-5 所示。

选中“生成测试套件”复选框，选择其他的选项后，单击“确定”按钮。NetBeans 集成开发环境将为包含的所有类生成测试类，并将它们存储在项目的“测试包”节点中。如果已经存在测试类，将会对这些类进行更新。生成的测试套件源代码如下所示。

Download at http://www.pin5i.com/


```
import junit.framework.*;
/*
 * RootSuite.java
 * JUnit based test
 *
 * Created on 2007年3月21日, 下午11:21
 */

/**
 *
 * @author Liu Bin
 */
public class RootSuite extends TestCase {

    public RootSuite(String testName) {
        super(testName);
    }

    protected void setUp() throws Exception {
    }

    protected void tearDown() throws Exception {
    }

    /**
     * 套件方法由 JUnit 模块自动生成
     */
    public static Test suite() {
        TestSuite suite = new TestSuite("RootSuite");
        suite.addTest(MyClassTest.suite());
        return suite;
    }

}
```

在“项目”窗口中，右键单击测试套件节点，选择弹出菜单的“运行文件”命令，可以执行测试套件。执行结果如图 8-6 所示。

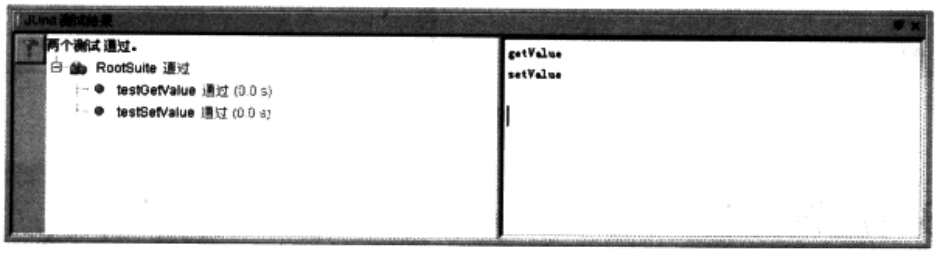


图 8-6 测试套件执行结果

说明：如果要调试 JUnit 测试，可以在“项目”窗口或者“文件”窗口中选择要调试的类，然后从主菜单中选择“运行”→“运行文件”→“调试 classname 的测试”命令（快捷键 Ctrl + Shift + F6）。NetBeans 集成开发环境将在调试器中运行测试，以检查测试是否有错误。

8.4 配置 JUnit

在 NetBeans 集成开发环境中，可以对 JUnit 进行配置，以定制自动生成的测试代码。选择主菜单中的“工具”→“选项”命令，打开“选项”对话框。单击“高级选项”按钮，打开“高级选项”对话框。单击“测试”→“JUnit 测试”节点，如图 8-7 所示。

Download at http://www.pin5i.com/

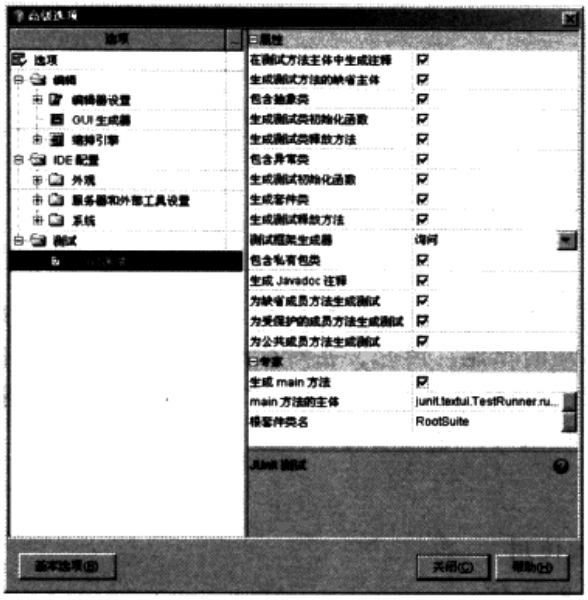


图 8-7 配置 JUnit

在这里可以定制 JUnit 测试的属性。例如，在“根套件类名”属性中修改创建的根测试套件的默认名称。

8.5 小结

本章主要介绍了如何在 NetBeans 集成开发环境中进行 JUnit 单元测试。在介绍的过程中已经假定了读者具有一定的 JUnit 测试基础知识，因此只对必要的 JUnit 知识进行了介绍。如果在学习本章的过程中，对 JUnit 有任何困惑，建议读者参考 JUnit 的相关知识。

通过本章的学习，读者应该掌握以下基本内容：

- 如何在 NetBeans 中创建空测试
- 如何根据现有的类创建测试
- 如何创建测试套件进行测试

Download at http://www.pin5i.com/

第9章 性能分析

在开发应用程序时，有时需要知道代码的执行效率如何以及内存的占用情况，以便于对代码进行优化、提高应用程序的性能。因此，在对代码进行优化前，对代码进行性能分析就显示十分必要。NetBeans 提供了一个优秀的代码分析工具 Profiler，本章将详细地介绍如何利用这个工具进行性能分析、监视应用程序的运行等。

本章主要内容

- Profiler 控制面板的使用
- 如何监视应用程序的运行
- 如何进行性能分析
- 如何分析内存的使用

本章亮点

- 快照的使用
- 自定义性能分析
- 在 NetBeans 中配置 Profiler

9.1 Profiler 概述

NetBeans Profiler 是一个旨在为 NetBeans 集成开发环境提供全面性能分析功能的项目。更多的项目信息以及技术文档，读者可以访问 http://profiler.netbeans.org。Profiler 可以无缝地集成到 NetBeans 集成开发环境中，帮助开发人员进行性能分析以及基本的 JVM 监视。利用这些功能，开发人员可以更有效地解决内存问题或与性能相关的问题。

9.1.1 安装 Profiler 工具

在 NetBeans 5.5 中，Profiler 安装包需要在 NetBeans 的官方网站 www.netbeans.org 或者 http://profiler.netbeans.org 下载并单独安装。在 NetBeans 6.0 中，图 9-1 所示的几个安装文件中都包含了 Profiler 工具。

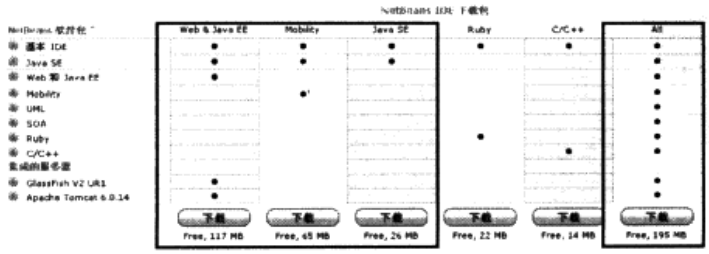


图 9-1 下载包含 Profiler 工具的安装程序

Download at http://www.pin5i.com/

启动 NetBeans 6.0 集成开发环境后，会在主菜单中出现一个 Profiler 菜单，如图 9-2 所示。Profiler 菜单中包含了进行性能分析的各种命令。

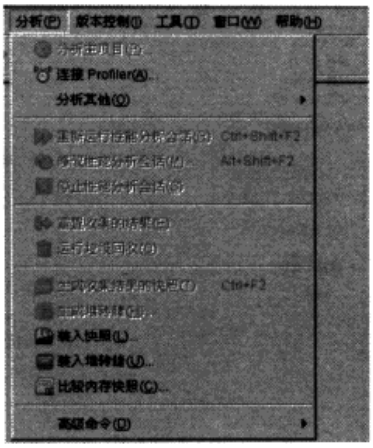


图 9-2 Profiler 菜单

9.1.2 校准 Profiler

对于每个用于性能分析的 JDK，Profiler 都需要执行一次初始校准。从校准中获取的数据将用于从收集的结果中排除额外产生的性能分析开销，以便能够提供更精确的执行时间信息。

校准数据与系统性能（CPU 速度、内存和总线吞吐量等）和 JDK 版本密切相关。在每次更改 JDK 或进行了如下的更改后，都应该运行校准以确保 Profiler 收集的结果准确无误。

- 任何硬件更改，例如更改 CPU、内存等。
- 操作系统的任何重大更改/升级。
- 用于性能分析的 Java 平台的升级。

在第一次调用 Profiler 时，系统会自动提示运行校准。如果本地计算机和 JVM 的校准数据不可用，系统也会作出提示。这时需要手动运行校准过程。选择 NetBeans 主菜单中的“分析”→“高级命令”→“运行 Profiler 校准”命令。打开“选择要校准的 Java 平台”对话框，如图 9-3 所示。

单击要校准的 JDK，然后单击“确定”按钮开始校准。校准完毕后，显示如图 9-4 所示的提示信息。

单击“显示详细信息”按钮，显示一些有关校准结果的基本信息，如图 9-5 所示。

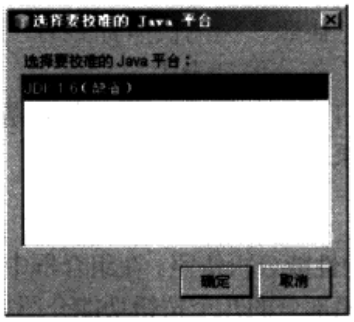


图 9-3 “选择要校准的 Java 平台”对话框

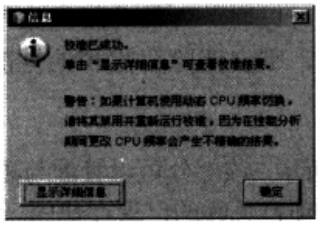


图 9-4 “信息”对话框



图 9-5 显示校准详细信息

注意：每个 JDK 的校准数据都保存在 NetBeans Home 目录的.nbprofile 子目录中。在不同的计算机或系统之间共享校准数据是没有任何意义的，只能导致分析结果的不可靠。

9.1.3 Profiler 控制面板

运行 Profiler 时，NetBeans 集成开发环境的左窗格中将显示 Profiler 控制面板。也可以手工打开这个窗口，在 NetBeans 的主菜单中选择“窗口”→“分析”→“Profiler 控制面板”命令。打开“Profiler 控制面板”窗口，如图 9-6 所示。“Profiler 控制面板”窗口中主要包括了控件、状态、性能分析结果、保存的快照、视图、基本遥测几个部分，下面详细介绍各个部分的主要功能。

1. 控件

控件部分如图 9-6 所示，在该部分包括几个按钮，用于执行不同的任务，例如执行性能分

Download at http://www.pin5i.com/

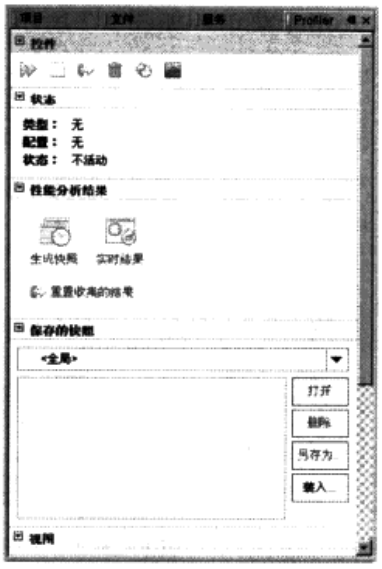


图 9-6 Profiler 控制面板 1

析、查看性能分析结果等。这几个按钮的功能说明如下。

- “重新运行上一次的性能分析”按钮：再次运行上一个性能分析命令。快捷键为 Ctrl + Shift + F2。
- “停止（终止）分析的应用程序”按钮：停止当前的性能分析命令。如果目标应用程序是由 Profiler 启动的，则还会停止该应用程序。
- “重置收集的结果缓冲区”按钮：把结果缓冲区清空。
- “运行垃圾回收”按钮：运行垃圾回收，回收无用的内存空间。
- “修改性能分析”按钮：打开“修改性能分析任务”对话框，可以在无须停止目标应用程序的情况下运行新的性能分析命令。
- “遥测概览”按钮：在“输出”窗口中打开“VM 遥测概览”窗口，可以显示较小的遥测图形版本。

2. 状态

状态部分如图 9-6 所示，在该部分显示当前被监视程序的类型、配置以及状态等信息。关于这部分的应用，读者可以参考第 9.2.1 节中的介绍。

3. 性能分析结果

性能分析结果部分如图 9-6 所示，该部分可以显示执行性能分析后的结果。单击“生成快照”按钮，可以在“源编辑器”窗口中打开此时的性能分析信息快照。关闭快照标签时，可以选择保存快照。保存快照时，快照名称将添加到“保存的快照”部分的列表中。

单击“实时结果”按钮，可以在“性能分析结果”标签中打开当前性能分析任务的结果。单击“重置收集的结果”按钮，则会清除已累积的性能分析结果。

4. 保存的快照

保存的快照部分如图 9-7 所示，该部分可以管理与项目关联的性能分析快照。在组合框中选择一个打开的项目时，将显示与该项目关联的已保存快照。保存的快照部分的几个按钮功能介绍如下。

- “打开”按钮：选中某个快照，单击该按钮可以打开快照窗口并显示快照内容。双击快照名称也可以打开这个快照。
- “删除”按钮：可以删除选中的快照。
- “另存为”按钮：可以把快照保存为另外的一个文件。
- “装入”按钮：从文件系统中载入一个文件。

5. 视图

视图部分如图 9-7 所示，该部分包含两个按钮，功能介绍如下：

- “VM 遥测”按钮：可打开“VM 遥测”窗口，显示有关虚拟机中线程活动、内存堆及垃圾回收的高级数据。
- “线程”按钮：可以打开“线程”窗口，显示应用程序线程活动。

6. 基本遥测

基本遥测部分如图 9-7 所示，该部分提供了有关当前性能分析任务的高级信息。可以通过单击“视图”部分中的“VM 遥测”按钮和“线程”按钮，查看该信息的图形表示。

Download at http://www.pin5i.com/

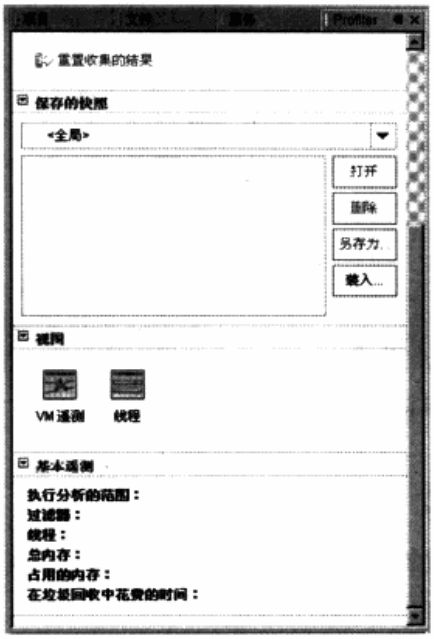


图 9-7 Profiler 控制面板 2

9.2 监视应用程序的运行

从分析内容上来讲，在 Profiler 中可以监视应用程序的运行、进行性能分析、对代码段进行分析、分析内存的使用以及进行自定义分析。

监视应用程序的运行可以确定应用程序方法使用的 CPU 时间，以及监视应用程序创建对象的过程，具体的监视内容包括：

- 堆内存大小；
- 垃圾回收统计信息；
- 线程计数；
- 线程的运行、休眠、等待、受阻等状态。

9.2.1 启动程序监视

如果要对主项目进行监视，可以在 NetBeans 主菜单中选择“分析”→“分析主项目”命令，打开“启用性能分析”对话框，如图 9-8 所示。

该对话框在第一次进行项目分析的时候出现。单击“确定”按钮，打开“性能分析任务”对话框。在该对话框中，单击“监视应用程序”按钮，显示如图 9-9 所示的界面。

说明：如果要对非主项目进行分析，可以首先在“项目”窗口中单击要监视的项目节点，然后在 NetBeans 的主菜单中选择“分析”→“分析其他”→“分析项目<选中的项目名称>”命令。

选中“启用线程监视”复选框，这样进行监视的时候会自动运行项目。单击“运行”按钮，启动主项目并进行性能分析。分析完毕后，会自动打开“VM 遥测概览”窗口、“线程”窗口以及 Profiler 窗口（Profiler 控制面板），下面分别进行介绍。

Download at http://www.pin5i.com/

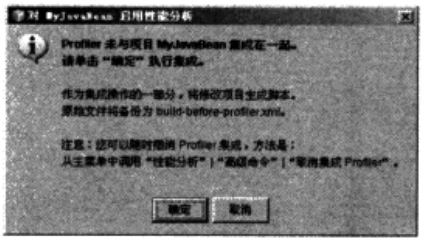


图 9-8 “启用性能分析”对话框

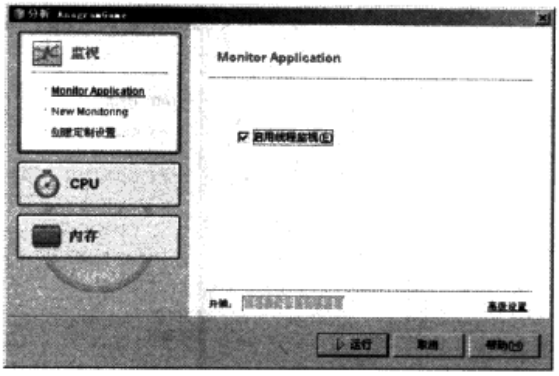


图 9-9 “性能分析任务”对话框



9.2.2 VM 遥测概览

程序刚启动时的“VM 遥测概览”窗口如图 9-10 所示，其中显示了有关堆内存的使用情况、虚拟机中线程活动以及垃圾回收的高级数据。

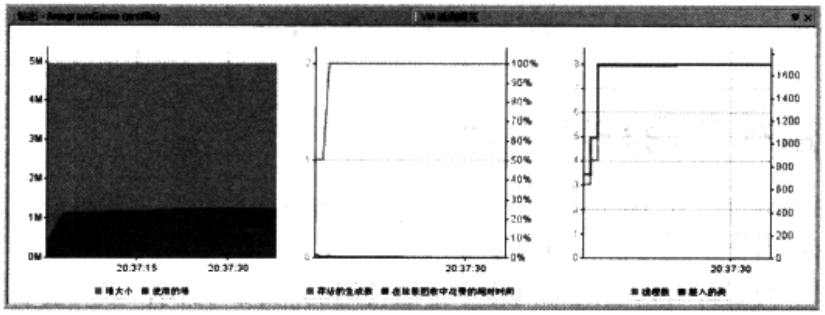


图 9-10 程序刚启动时的“VM 遥测概览”窗口

最左边的一个图显示了当前的堆内存数量，以及被监视程序使用的内存数量。红色的阴影部分表示分配的 Java 虚拟机堆大小，紫色的覆盖部分表示实际使用的堆空间大小。这些数值也可以在 Java 应用程序中通过使用如下两个方法获得。

- java.lang.Runtime.totalMemory()：获得总的堆内存数量。
- java.lang.Runtime.freeMemory()：获得空闲的堆内存数量。

中间的图显示了垃圾回收（Garbage Collection, GC）中花费的相对时间和存活年代数。■粉颜色的线表示存活的生成数，显示从连接 Profiler 那一刻起，在 Java 虚拟机堆上分配的所有对象的不同“生存期”数量。该颜色的线是以图形左侧的 Y 轴为参照进行绘制的。



说明：“生存期”是对象存活时的垃圾回收次数。

■ 紫颜色的线表示在垃圾回收中花费的相对时间，该时间是以目标 Java 虚拟机在运行垃圾回收（并暂停所有应用程序线程）时所花费的时间占总执行时间的百分比来测量的。该颜色的线以图形右侧上的 Y 轴为参照绘制。如果蓝线占据较大的百分比，则需要考虑调整 Java 虚拟机，配置更大的堆。

最右边的一个图显示应用程序 Java 虚拟机中的线程总数。将鼠标移动到该图形上时，会显示当前线程数和最大线程数。

在被监控的应用程序中进行一些操作，此时的“VM 遥测概览”窗口如图 9-11 所示。可以看到使用的内存、存活的生成数都发生了变化。

Download at http://www.pin5i.com/

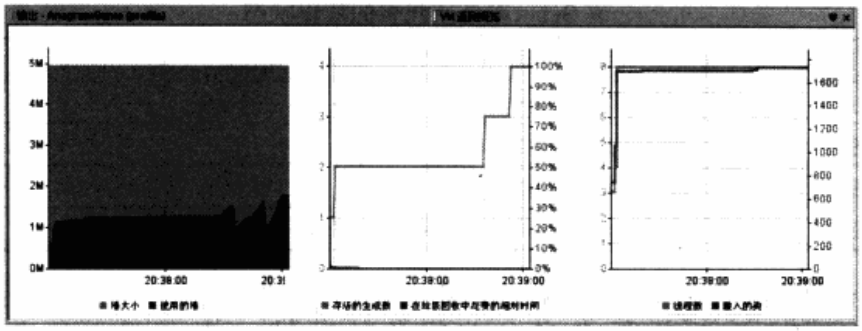


图 9-11 程序启动后执行一些操作后“VM 遥测概览”窗口



说明：在 NetBeans 主菜单中选择“分析”→“视图”→“VM 遥测”命令，可以打开“VM 遥测”窗口，在该窗口中用三个标签分别显示“VM 遥测概览”窗口中的三个图形。



9.2.3 “线程”窗口

“线程”窗口显示了有关应用程序线程活动的详细信息，可以用来监视线程活动，如图 9-12 所示。

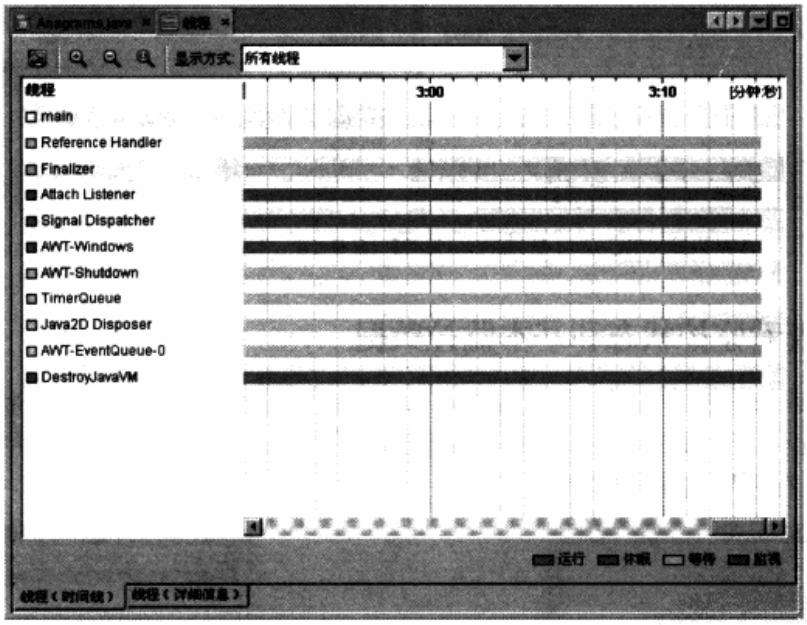


图 9-12 “线程”窗口的“线程（时间线）”标签

在监视应用程序时，“线程”窗口会自动打开。也可以通过如下的两种方式，手工打开“线程”窗口：

- 在 NetBeans 主菜单中选择“分析”→“视图”→“线程”命令；
- 在 Profiler 控制面板中单击“线程”按钮。

在“线程”窗口的“线程（时间线）”标签中，横向为程序运行的时间，纵向列出了当前程序中使用的所有线程以及线程的状态。线程状态使用颜色编码来区分，不同颜色的具体含义如表 9-1 所示。

Download at http://www.pin5i.com/

表 9-1 颜色所代表的线程状态

颜 色	线程状态
绿色	线程正在运行或准备运行
紫色	线程处于休眠状态
黄色	线程处于等待状态
红色	线程在尝试访问同步的块或方法时受阻

如果此时执行一些界面操作，会显示如图 9-13 所示的界面。对比图 9-13 和图 9-12，可以发现 AWT-EventQueue-0 线程中多一些绿色的区域，表示界面操作的确影响了 AWT-EventQueue-0 线程。

右键单击 AWT-EventQueue-0 线程，然后选择弹出菜单中的“显示详细信息”命令，打开“线程（详细信息）”标签，如图 9-14 所示。在饼图中显示了每种状态所花的时间。

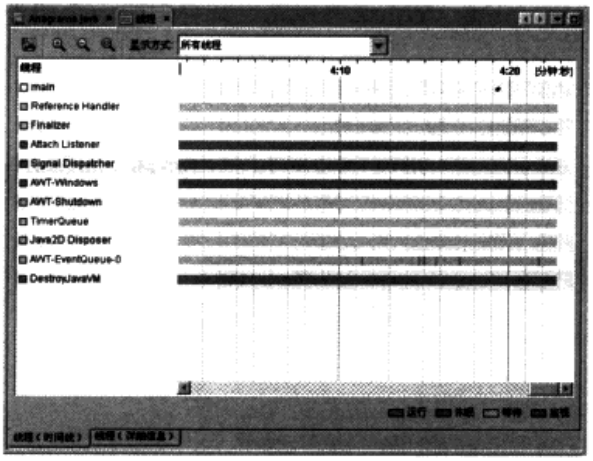


图 9-13 界面操作对 AWT-EventQueue-0 线程的影响

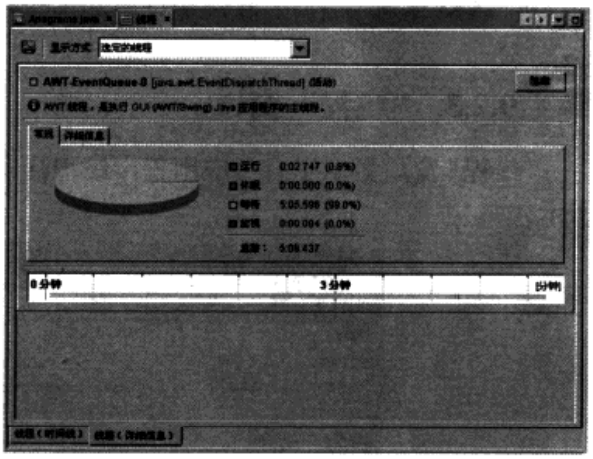


图 9-14 “线程”窗口的“线程（详细信息）”标签

9.2.4 查看监控状态和基本监控数据

在 Profiler 进入监控状态后，可以在 Profiler 控制面板中读取程序的一些状态。在如图 9-15 所示的界面中显示了当前性能分析的类型及状态。在如图 9-16 所示的界面中显示了当前程序的线程数量、内存使用情况等信息。



图 9-15 监控的类型及状态



图 9-16 当前程序的线程数量及内存使用情况

在分析结束后，关闭应用程序时会显示如图 9-17 所示的对话框，单击“确定”按钮即可退出监视状态。如果不希望该对话框再次出现，那么可以选中“不要再次显示此消息”复选框。

Download at http://www.pin5i.com/

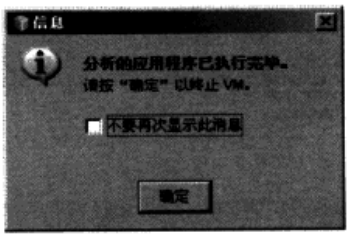


图 9-17 退出应用程序监视时的提示对话框

9.3 进行 CPU 性能分析

上一节讲的是对项目进行监视，这一节将介绍如何对应用程序的性能进行分析。在性能分析后会给出执行时间最长的方法，这样在进行性能优化的时候，可以对比较花时间的方法进行有的放矢的改进。

9.3.1 启动性能分析

按照上一节介绍的方法，打开“性能分析任务”对话框，并单击 CPU 按钮，显示界面如图 9-18 所示。

选中“整个应用程序”单选按钮，可以对整个应用程序进行性能分析。选中“部分应用程序”单选按钮，可以对部分程序进行分析。

在“过滤器”下拉列表框中可以选择必要的过滤器。单击“显示过滤器值”超链接，会打开如图 9-19 所示的过滤值对话框。性能分析器会对列表中包含的类的方法进行性能分析。

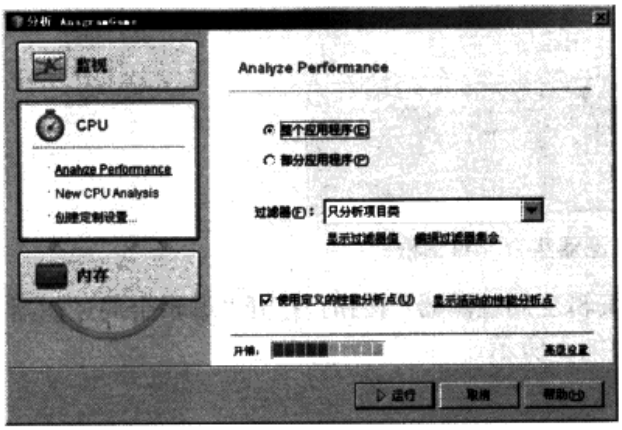


图 9-18 CPU 性能分析

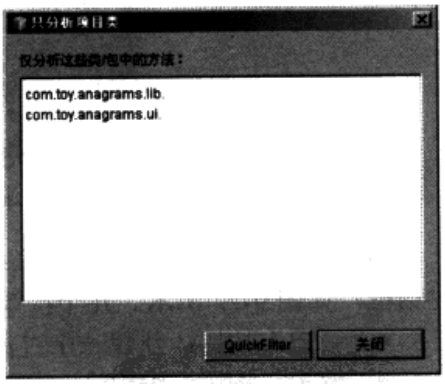


图 9-19 过滤值对话框

单击“QuickFilter”按钮，打开如图 9-20 所示的“设置快速过滤器”对话框。选中“排除”单选按钮，则下面列表中的类会在进行分析时被排除；选中“包含”单选按钮，则只会对下面列表中的类进行分析。

在图 9-18 所示的对话框中，单击“编辑过滤器集合”超链接，可以打开如图 9-21 所示的“定制过滤器集合”对话框，如图 9-21 所示。

单击“添加”按钮，可以添加一个新的过滤器集合。单击“删除”按钮可以删除选中的过滤器集合。对于某个过滤器集合，可以在“过滤器集合名称”文本框中修改过滤器的名字。如果选择“排除”单选按钮，那么会排除“全局过滤器”列表中选中的全局过滤器；如果选中“包含”单选按钮，那么会包含“全局过滤器”列表中选中的全局过滤器。

Download at http://www.pin5i.com/

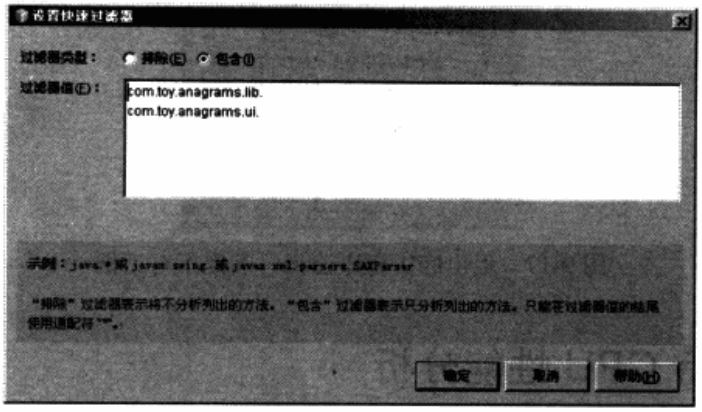


图 9-20 “设置快速过滤器”对话框

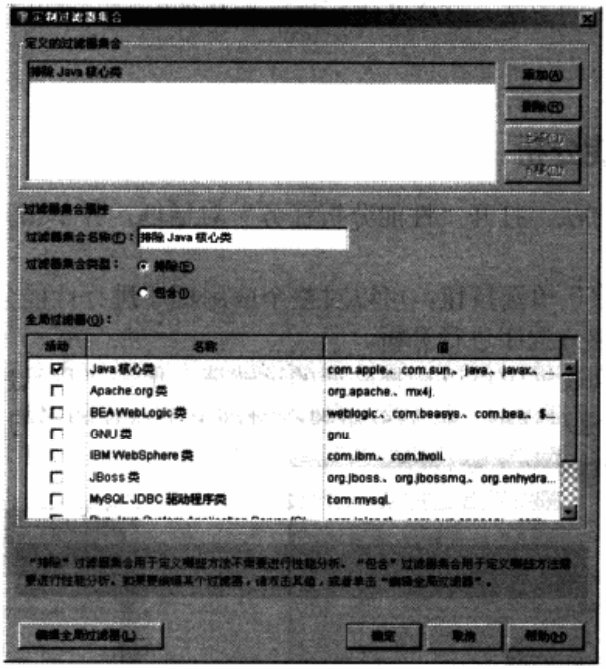


图 9-21 “定制过滤器集合”对话框

如果要定制全局过滤器，那么可以单击“编辑全局过滤器”按钮，打开“编辑全局过滤器”对话框，如图 9-22 所示。几个按钮的功能说明如下所示。

- “新建”按钮：新建一个过滤器。
- “编辑”按钮：编辑选中的过滤器。
- “删除”按钮：删除选中的过滤器。
- “上移”按钮：把选中的过滤器向上移动一个位置。
- “下移”按钮：把选中的过滤器向下移动一个位置。

选择完毕后，返回“分析”对话框。单击“运行”按钮，开始进行性能分析。



9.3.2 查看快照

在 Profiler 控制面板中单击“生成快照”按钮，或者在 NetBeans 的主菜单中选择“分析”→“生成收集结果的快照”命令，打开 CPU 快照窗口，如图 9-23 所示。在该窗口中会显示有关方法调用树、时间和调用次数的数据。

Download at http://www.pin5i.com/

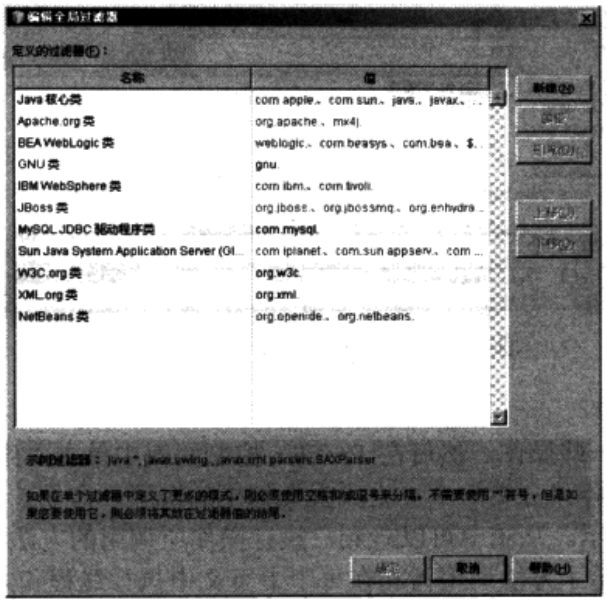


图 9-22 编辑全局过滤器

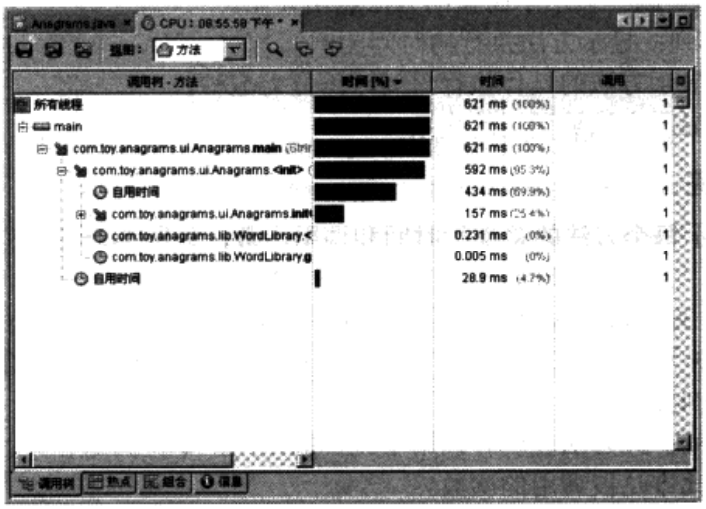


图 9-23 CPU 快照窗口

1. 快照窗口的工具栏

在 CPU 快照窗口的工具栏中，单击“将快照保存到项目中”按钮，可以把当前的快照保存的项目中。

单击“将快照保存到定制文件夹中”按钮，会打开如图 9-24 所示的“选择文件或目录”对话框，用于选择保存快照文件的目录并输入文件名。在默认情况下，快照文件的后缀为.nps。

在图 9-23 中，单击工具栏上“保存图像的当前视图”按钮，可以把快照作为图像进行保存。工具栏上的“视图”下拉列表框用于控制结果视图的粗糙度，可以选择如下几个选项。

- “方法”：方法级别视图，根据全限定方法名称来显示结果。该视图为默认视图类型。
- “类”：类级别视图，同一类所有方法的结果都聚集在一个条目中。
- “包”：包级别视图，属于同一包的类的所有方法都聚集在一起。如果有大量的方法要进行分析，并且需要大致了解应用程序中哪一段代码的执行时间最长，此视图可能特别有用。

单击工具栏中的“结果中查找”按钮，可以在生成的结果中进行搜索。

Download at http://www.pin5i.com/

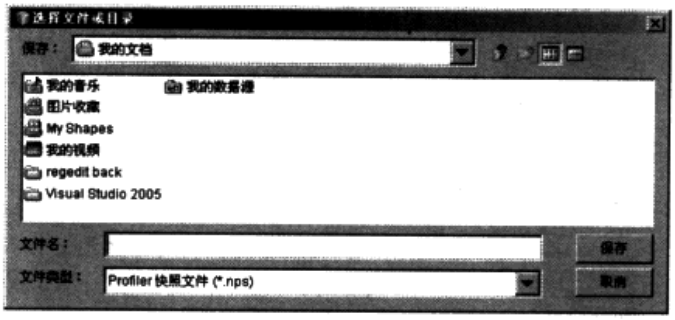


图 9-24 “选择文件或目录”对话框

2.“调用树”标签

在应用程序中执行一些操作，然后在 Profiler 控制面板中单击“生成快照”按钮，或者在 NetBeans 的主菜单中选择“分析”→“生成收集结果的快照”命令，会打开另外的一个 CPU 快照窗口，如图 9-23 所示。从图中可以看到，执行操作所调用的方法的具体执行时间。

“调用树”标签中显示了方法调用树和在每个上下文中执行线程和方法的时间/调用次数（上下文是指向方法调用的唯一方法调用树）。该标签中几个列的含义如下所示。

- “调用树–方法”列：显示了树型的方法调用关系。
- “时间［%］”列：显示方法执行的时间比例。
- “时间”列：显示方法的具体执行时间，单位为毫秒。
- “次数”列：显示方法被调用的次数。

3.“热点”标签

“热点”标签显示每个方法的总执行时间和调用次数，如图 9-25 所示。该标签中几个列的含义如下所示。

- “热点–方法”列：按照执行时间的大小排序的方法列表，即热点方法列表。
- “自用时间［%］”列：在此方法的所有调用中花费的时间比例。此值越大，表示方法单次执行花费的时间越多。
- “自用时间”列：此方法在所有调用中花费的时间，单位为毫秒。
- “调用”列：此方法被调用的次数。

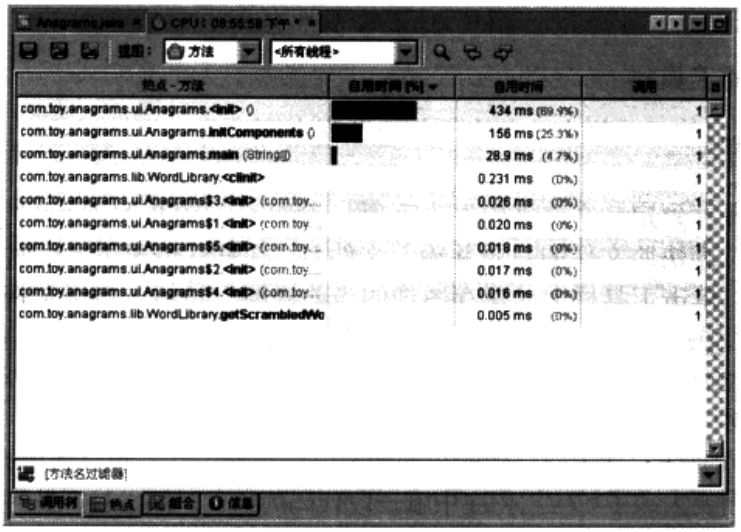


图 9-25 “热点”标签

Download at http://www.pin5i.com/

技巧：在进行性能优化的时候，可以重点对一些花时间比较多或者被调用次数比较多的方法进行研究。一旦找到改进方法，将会对系统性能的提升意义重大。

4. “组合”标签

“组合”标签在窗口的上半部分显示调用树信息并在窗口的下半部分显示热点数据，如图9-26所示。这个标签中各个内容的含义与“调用树”标签和“热点”标签一样，这里不再赘述。在这个标签中可以对方法的执行情况进行综合的比较。

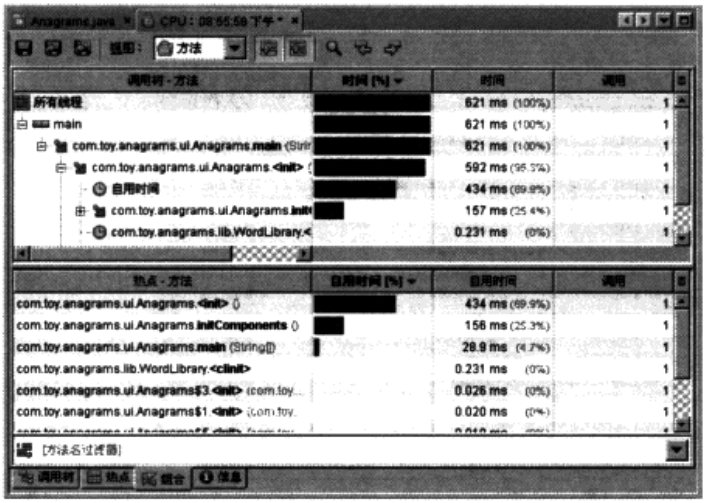


图9-26 “组合”标签

5. “信息”标签

“信息”标签显示有关生成快照的时间、保存位置以及使用的性能分析设置的数据，如图9-27所示。在该标签中显示了快照的抓取时间、性能分析类型、CPU性能分析类型、CPU记时器、分析过滤器、是否分析方法调用、是否分析线程、是否分析getter和setter方法、是否分析空方法等内容。

6. 快照的保存和读取

通过快照窗口可以把快照保存在项目中，或者任意指定的文件夹中。如果把快照保存在项目中，可以在如图9-28所示的Profiler控制面板中显示保存的快照，并可以在任意时候打开浏览。

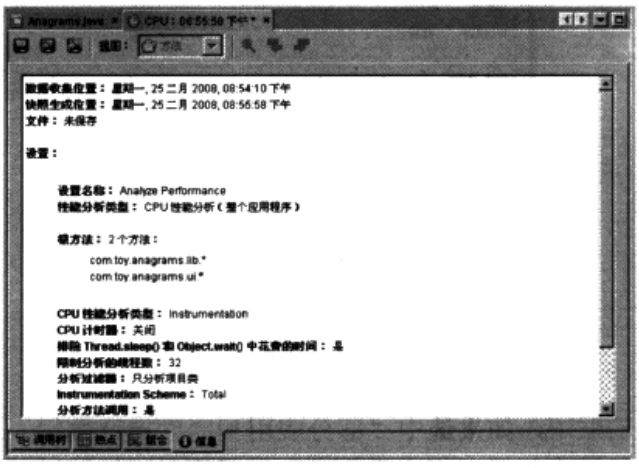


图9-27 “信息”标签



图9-28 在Profiler控制面板中读取和载入快照

Download at http://www.pin5i.com/

单击“装入”按钮，可以从文件系统中打开快照文件（*.nps）。另外，在 NetBeans 主菜单中选择”分析”→“装入快照”命令，也可以打开一个快照文件。

9.3.3 查看性能分析结果

在进行性能分析时，如果要查看性能分析的实时数据而不是快照，可以在 NetBeans 主菜单中选择“窗口”→“性能分析”→“实时结果”命令，打开“即时性能分析结果”窗口，如图 9-29 所示。

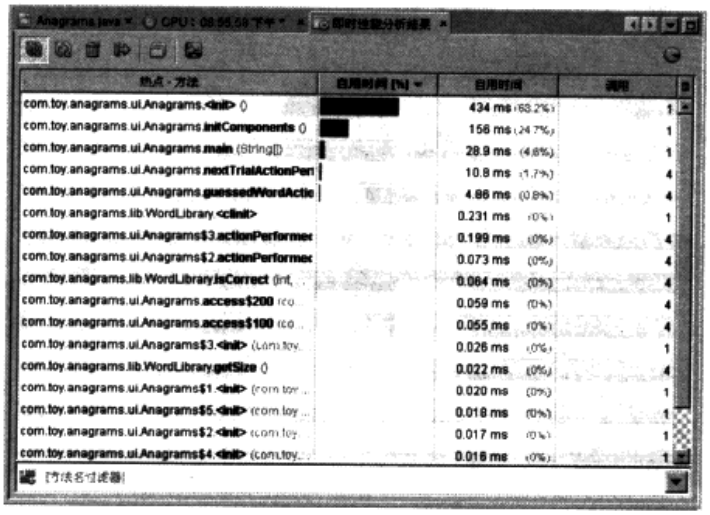


图 9-29 “即时性能分析结果”窗口

9.3.4 查看性能分析状态和基本数据

在 Profiler 进入性能分析状态后，可以在 Profiler 控制面板中读取性能分析的一些状态和数据。在如图 9-30 所示的界面中显示了当前性能分析的类型及状态。在如图 9-31 所示的界面中显示了当前分析的范围、过滤器、线程数量、内存使用情况等信息。

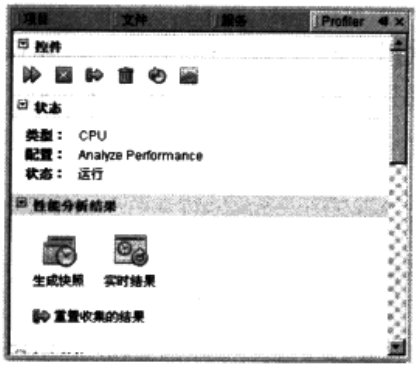


图 9-30 性能分析的类型及状态

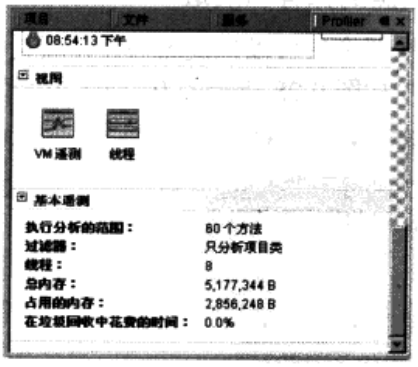


图 9-31 基本性能分析数据

9.3.5 指定根方法

在如图 9-18 所示的“性能分析任务”对话框中，如果选中“部分应用程序”单选按钮，会在后面出现一个“定义”超链接。单击该超链接会打开“指定根方法”对话框，选择要进行分析的根方法，如图 9-32 所示。

Download at http://www.pin5i.com/

单击“手动添加”按钮，可以打开“定义新的根方法”对话框，如图 9-33 所示。在这个对话框中，用户需要手工输入根方法。在“类名”文本框中输入要包含根方法的类名。在“方法名称”文本框中输入要进行性能分析的方法。在“方法 VM 签名”文本框中为方法提供虚拟机格式的方法签名。

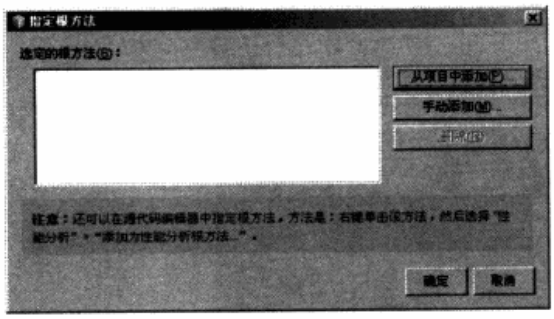


图 9-32 “指定根方法”对话框

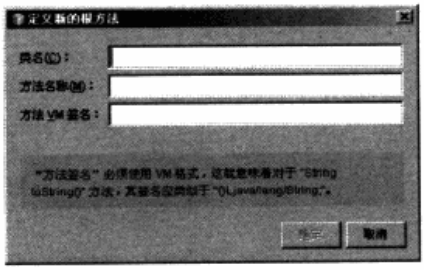


图 9-33 “定义新的根方法”对话框

如果在如图 9-32 所示的对话框中单击“从项目中添加”按钮，会打开如图 9-34 所示的“选择根方法”对话框。

在列表中需要选中要进行分析的方法名称。选中“显示所有项目”复选框则会显示所有打开的项目，而不只是要进行性能分析的项目。

在源代码编辑器中，如果选中某个方法的名字，然后单击鼠标右键。选择弹出菜单中的“性能分析”→“添加性能分析根方法”命令，则会打开“选择设置配置”对话框，如图 9-35 所示。在这个对话框中，可以选择根方法应用的配置。选中某个配置，单击“确定”按钮，就会添加一个根方法了。

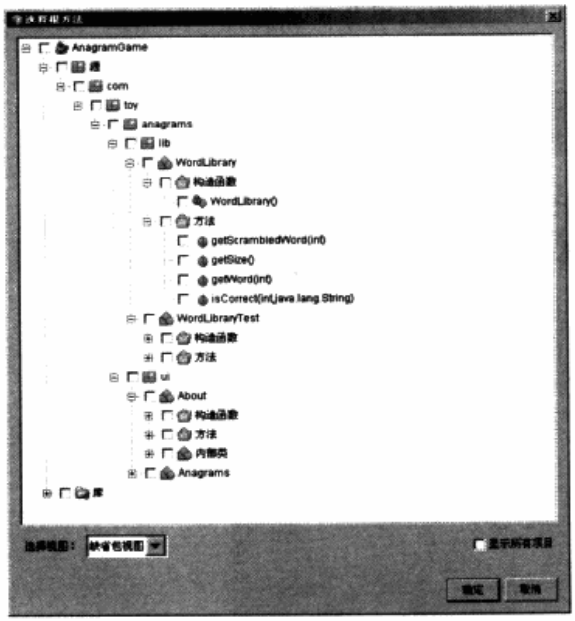


图 9-34 “选择根方法”对话框

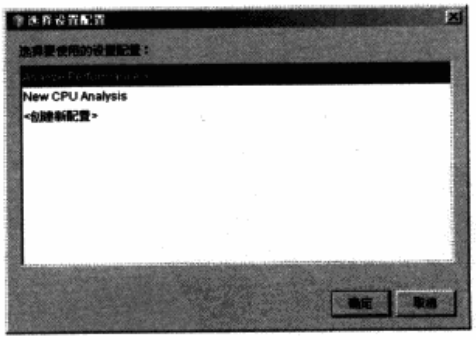


图 9-35 “选择设置配置”对话框

9.4 使用分析点

在源代码要插入分析点的行上单击鼠标右键，选择弹出菜单中的“性能分析”→“插入性能分析点”命令，打开“新建性能分析点”对话框，如图 9-36 所示。

Download at http://www.pin5i.com/

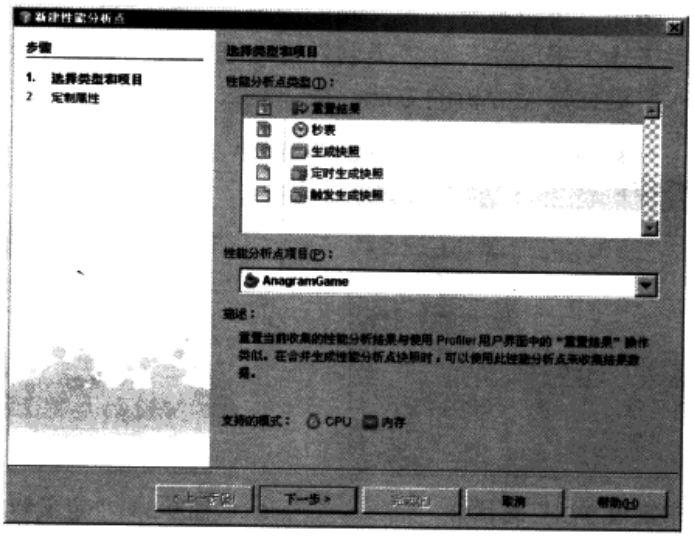


图 9-36 “新建性能分析点”对话框

在“性能分析点类型”列表中选择某一个类型，单击“下一步”按钮，即可创建不同类型的分析点。

9.4.1 创建重置结果类型分析点

重置结果类型分析点可以用来重置收集的分析结果。当和快照类型分析点结合使用时，可以用来获得收集结果的变化。在“新建性能分析点”对话框的“性能分析点类型”列表中，选择“重置结果”节点，单击“下一步”按钮打开“定制重置结果分析点属性”对话框，如图 9-37 所示。

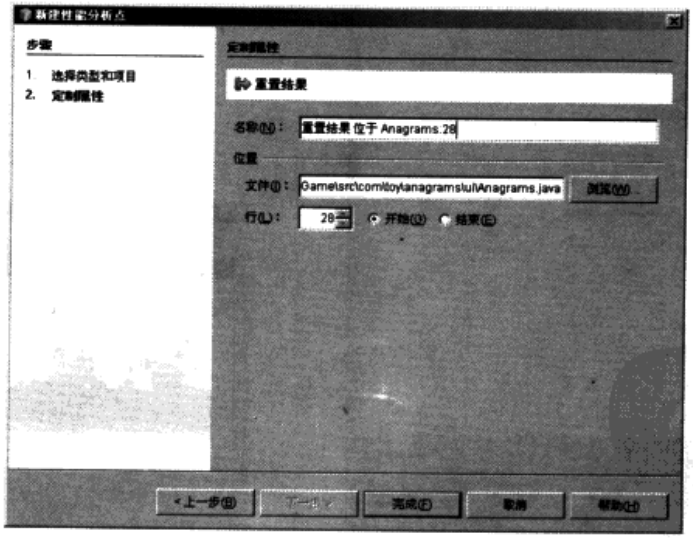


图 9-37 “定制重置结果分析点属性”对话框

在“名称”文本框中输入分析点的名字，后面的数字表示分析点所在的行。在“文件”文本框中给出分析点所在的文件路径。在“行”微调编辑框中修改分析点所在的行。选中“开始”单选按钮，表示该行是分析点的起始行；选中“结束”单选按钮，表示该行是分析点的结束行。单击“完成”按钮，即可创建一个重置结果类型分析点。

Download at http://www.pin5i.com/

9.4.2 创建秒表类型分析点

秒表类型分析点可以用来报告代码片段运行时的时间或者其执行时间。在“新建性能分析点”对话框的“性能分析点类型”列表中，选择“秒表”节点，单击“下一步”按钮打开“定制秒表分析点属性”对话框，如图 9-38 所示。

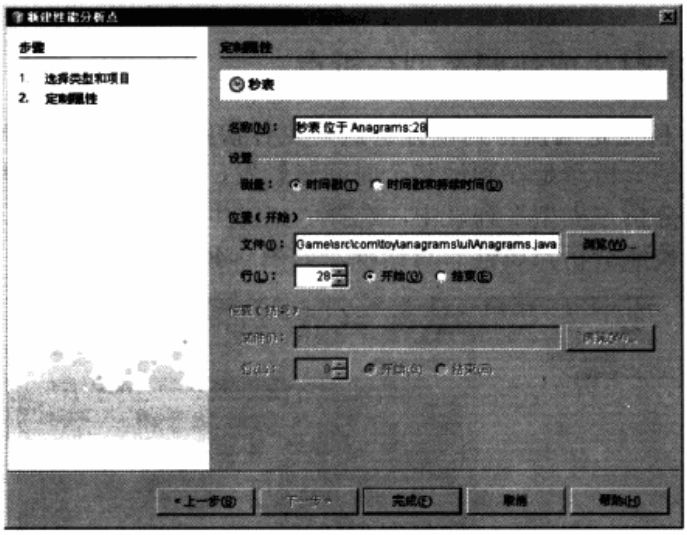


图 9-38 “定制秒表分析点属性”对话框

如果选中“时间戳”单选按钮，则会报告执行代码时的时间；如果选中“时间戳和持续时间”单选按钮，则会报告代码执行时的时间以及代码执行的时间。对话框中其他参数的说明，请参考图 9-37 的说明。

9.4.3 创建生成快照类型分析点

当执行到快生成照类型分析点时，就会自动为当前收集到的分析结果进行一次快照。在“新建性能分析点”对话框的“性能分析点类型”列表中，选择“生成快照”节点，单击“下一步”按钮打开“生成定制快照分析点属性”对话框，如图 9-39 所示。

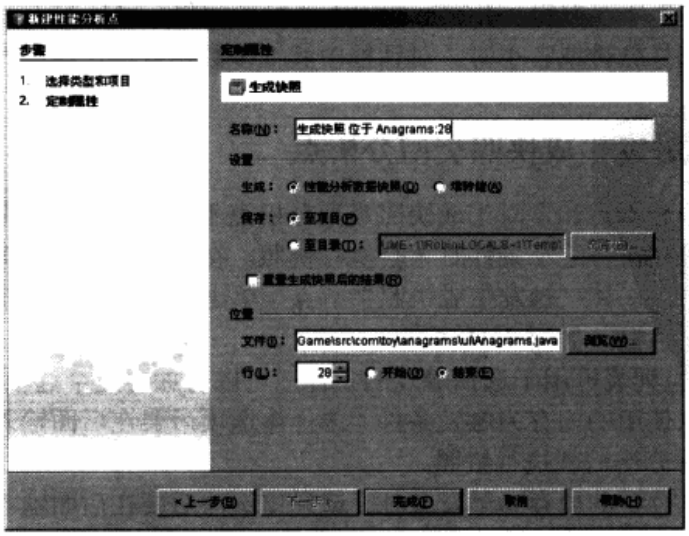


图 9-39 “生成定制快照分析点属性”对话框

Download at http://www.pin5i.com/

选中“性能分析数据快照”单选按钮，则会对收集的分析结果进行快照；选中“堆转储”单选按钮，则会对内存执行快照。

选中“至项目”单选按钮，则快照会保存在项目中；选中“至目录”单选按钮，快照会保存在指定的目录中。

选中“重置生成快照后的结果”复选框，在执行快照操作后会重置收集的分析结果。对话框中其他参数的说明，请参考图 9-37 的说明。



9.4.4 创建定时生成快照类型分析点

生成快照类型分析点是当执行到该分析点时执行快照操作，快照的执行依赖于某个代码行。定时生成快照类型分析点是在到达某个时间时触发，仅仅依赖于时间。在“新建性能分析点”对话框的“性能分析点类型”列表中，选择“定时生成快照”节点，单击“下一步”按钮打开“定制定时生成快照分析点属性”对话框，如图 9-40 所示。

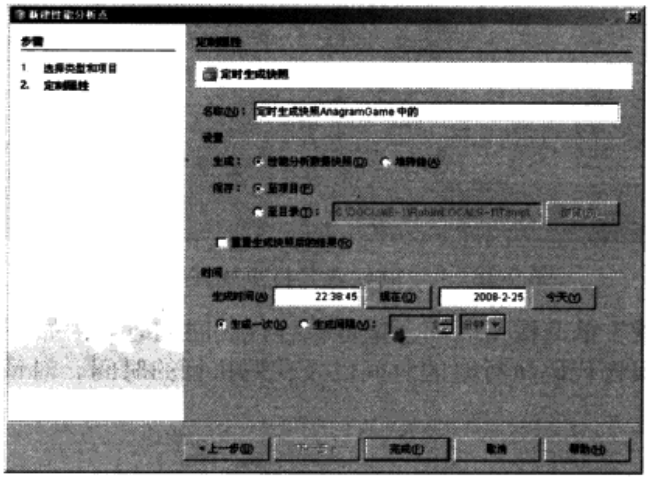


图 9-40 “定制定时生成快照分析点属性”对话框

“生成时间”文本框用于输入希望执行快照的时间，单击“现在”按钮可以自动填写当前的时间。后面的文本框用于输入执行快照的日期，单击“今天”按钮，则会自动填写当天的日期。

选中“生成一次”单选按钮，则只会执行一次快照。如果希望以一定时间间隔重复执行，则要选中“生成间隔”单选按钮，并在后面输入执行的间隔。最后的下拉列表框用于设置时间间隔的单位，可以选择分钟或者小时。对话框中其他参数的说明，请参考图 9-39 的说明。



9.4.5 创建触发生成快照类型分析点

与生成快照类型分析点和定时生成快照类型分析点不同，触发生成快照类型分析点依赖于某个触发条件，只有在符合某个条件时才会执行快照。在“新建性能分析点”对话框的“性能分析点类型”列表中，选择“触发生成快照”节点，单击“下一步”按钮打开“定制触发生成快照分析点属性”对话框，如图 9-41 所示。

“生成条件”下拉列表框用于选择触发的条件，可以有如下几个选项。

- 使用的堆：以使用的内存为触发条件。选择该选项后要在后面输入一个百分比，即内存使用达到该百分比后就执行快照。
- 分配的堆：以分配的内存为触发条件。选择该选项后要在后面输入一个内存的大小（单位为 MB），即内存分配达到该数值后就执行快照。

Download at http://www.pin5i.com/

- 存活的年代数：以存活的年代为触发条件。选择该选项后要在后面输入一个数值，即存活的年代超过该值后就执行快照。
- 装入的类：以装载的类的个数为触发条件。选择该选项后要在后面输入一个数值，即装载的类的个数达到该值后就执行快照。

选中“生成一次”单选按钮，则只会执行一次快照（第一次满足触发条件时）；选中“每次都生成”单选按钮，则在每次满足触发条件时，都会执行快照。对话框中其他参数的说明，请参考图 9-39 的说明。

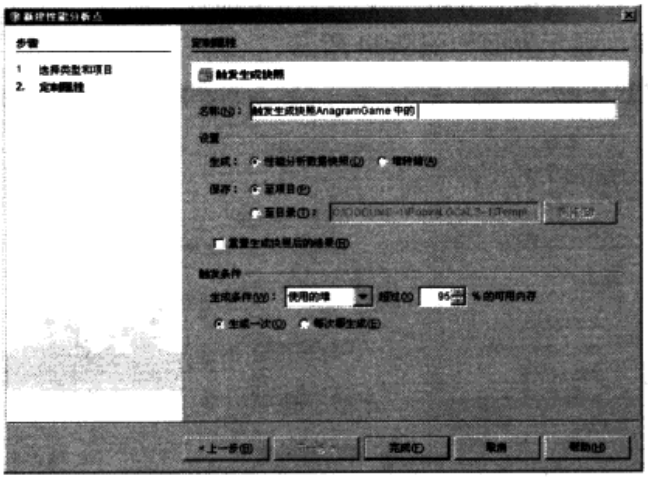


图 9-41 “定制触发生成快照分析点属性”对话框

9.4.6 使用分析点

在 NetBeans 主菜单的中选择“分析”→“分析主项目”命令，打开“性能分析任务”对话框。在该对话框中，单击“CPU”按钮，显示如图 9-42 所示的界面。要使用分析点，就需要选中“使用定义的性能分析点”复选框。单击后面的“显示活动的性能分析点”超链接，会打开“活动的分析点”对话框，如图 9-43 所示。在该对话框中可以查看项目中已经插入的不同类型的分析点。

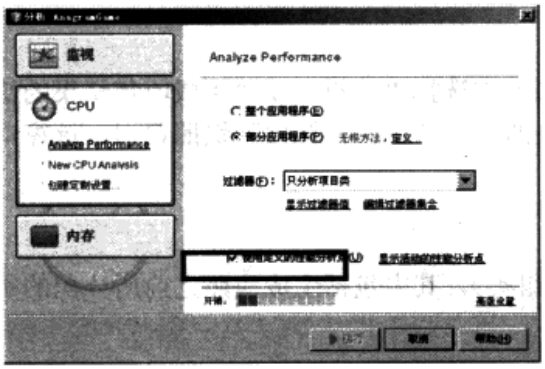


图 9-42 使用分析点

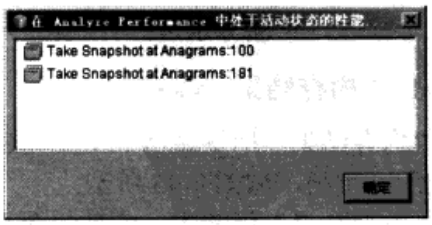


图 9-43 “活动的分析点”对话框

9.5 分析内存的使用

对内存的使用进行分析，可以帮助开发人员了解内存的使用情况，这样可以对内存的使用进行优化。例如，通过分析后，发现某个对象占有大量的内存。这样就可以对这个对象进行优

Download at http://www.pin5i.com/

化，以节省内存。

在开发过程中，有时会出现内存的泄漏，即某个对象已经不再被使用了，但是由于仍然保存着对这个的对象的引用，从而使得这个对象继续占据着内存资源。进行内存分析可以使得开发人员轻松地发现这些问题，从而对程序进行改进。



9.5.1 启动内存分析

在 NetBeans 主菜单中选择“分析”→“分析主项目”命令，打开“性能分析任务”对话框。单击“内存”按钮，显示界面如图 9-44 所示。

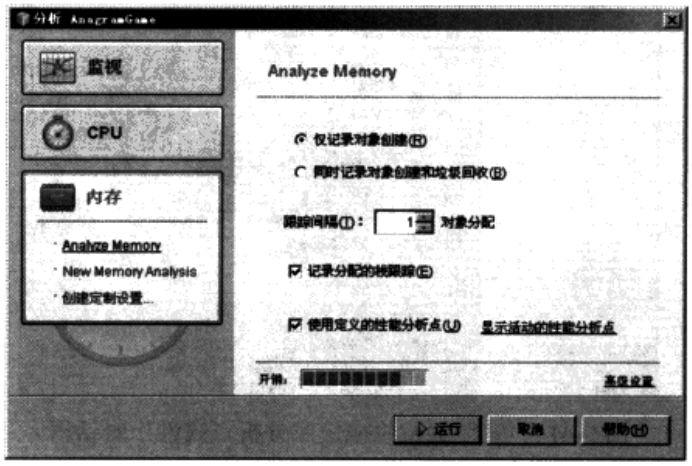


图 9-44 “性能分析任务”对话框

选中“仅记录对象创建”单选按钮，则只对对象的创建进行记录和分析。选中“同时记录对象创建和垃圾回收”单选按钮，则除了记录对象实例的分配情况，还会记录它们是否被垃圾回收。

选择“记录分配的栈跟踪”复选框，内存性能分析结果快照中将包含附加信息，通过查看使用这些信息可以获得方法调用的反向调用树。单击“运行”按钮，开始进行内存分析。



9.5.2 内存快照

在进入内存分析状态后，在 NetBeans 主菜单中选择“分析”→“生成收集结果的快照”命令，或者在 Profiler 控制面板中单击“生成快照”命令，会打开内存快照窗口。该窗口中包含三个标签，下面分别进行介绍。

1.“内存结果”标签

“内存结果”标签显示一个类列表（包括数组类），其中包括生成快照之前所分配的实例总数及所占空间，如图 9-45 所示。

“内存结果”标签中，各列的含义介绍如下。

- 分配的字节（图形）：以图形的形式显示活动实例占用的字节数。
- 分配的字节（数字）：显示活动实例占用的内存字节数以及其占用的百分比。
- 分配的对象：自应用程序启动或者上次结果重置后，创建的类的所有实例数。

在此标签中，可以右键单击任何类，然后选择弹出菜单中的“转至源代码”命令，会在源代码编辑器中打开对应的代码。

Download at http://www.pin5i.com/

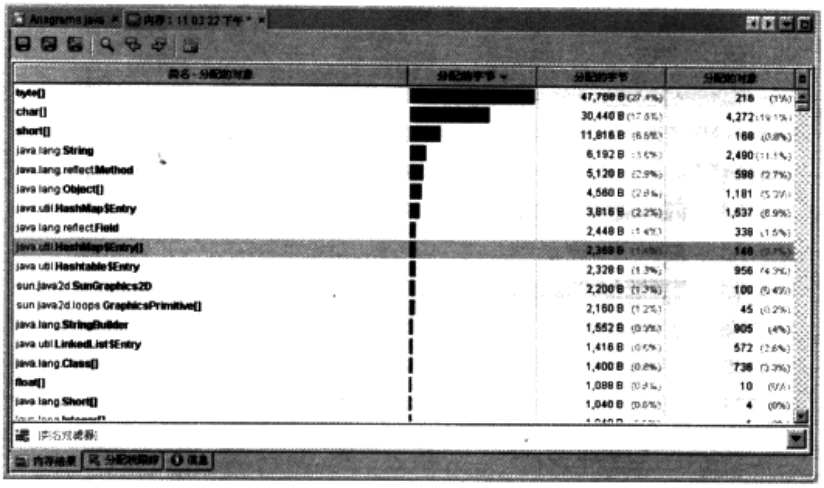


图 9-45 “内存结果”标签



说明：在 NetBeans 主菜单中选择“分析”→“视图”→“实时结果”命令，或者在 Profiler 控制面板中单击“实时结果”命令，会打开“性能分析结果”窗口，显示内存的使用情况。由于这个窗口和“内存结果”标签差不多，这里不再赘述。

2.“分配栈跟踪”标签

“分配栈跟踪”标签可以显示一个反向调用图形及其指向给定类对象分配的所有调用路径，如图 9-46 所示。

在“内存结果”标签中，可以右键单击任何类，然后选择弹出菜单中的“显示分配栈跟踪”命令，会在“分配栈跟踪”标签中显示选定类的栈跟踪。



注意：只有在如图 9-44 所示的“性能分析任务”对话框中勾选了“记录分配的栈跟踪”复选框后，才能使用此标签。

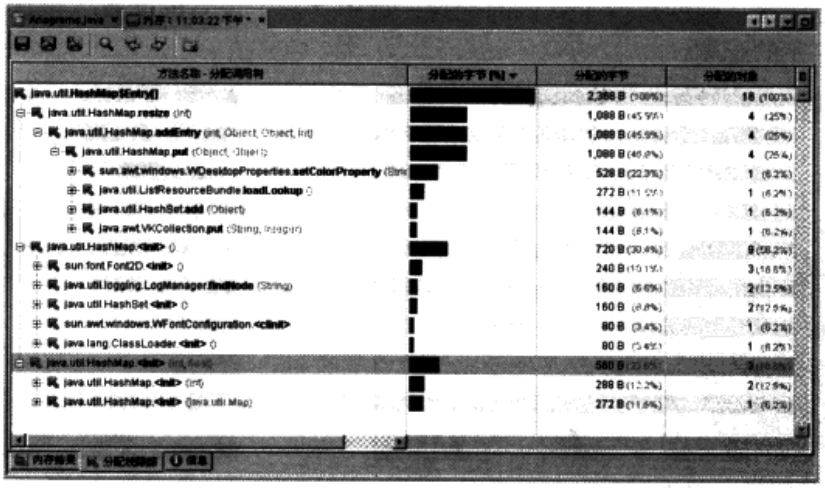


图 9-46 “分配栈跟踪”标签

3.“信息”标签

“信息”标签显示有关生成快照的时间、保存位置、是否记录了栈跟踪、是否在获取结果时运行垃圾回收等信息，如图 9-47 所示。

Download at http://www.pin5i.com/

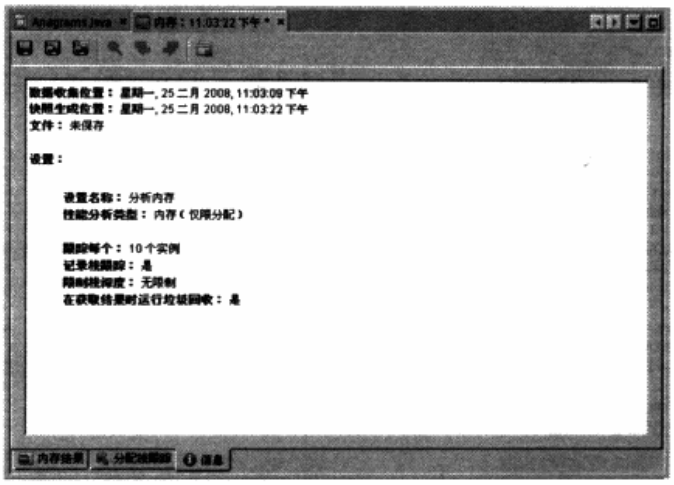


图 9-47 “信息”标签

9.6 自定义性能分析

前面几节介绍了几种标准的分析方法，本节将介绍如何创建自己的性能分析配置，即自定义性能分析。自定义性能分析给开发人员提供了最大的灵活性，可以根据需要对分析的各种参数进行设置。

9.6.1 创建定制的性能分析配置

在 NetBeans 主菜单的中选择“分析”→“分析主项目”命令，打开“性能分析任务”对话框。无论是选择监视、CPU 还是内存都包含一个“创建定制设置”超链接，如图 9-48 所示。

单击“创建定制设置”按钮，会打开“新建定制配置”对话框，如图 9-49 所示。最上面的三个单选按钮分别用于选择配置文件的类型。在“名称”文本框中可以输入配置的名字。选中“默认”单选按钮，则创建一个默认的配置；选中“基于现有的配置”单选按钮，可以选择使用一个已经存在的配置。

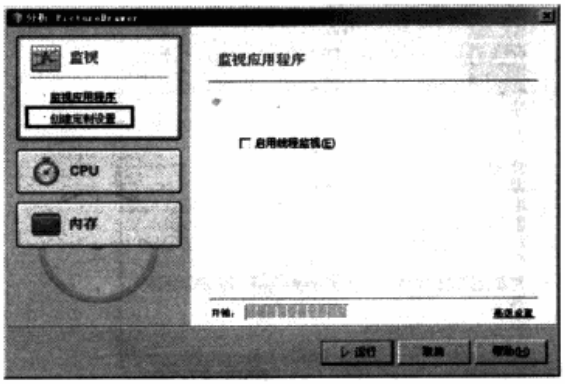


图 9-48 “性能分析任务”对话框

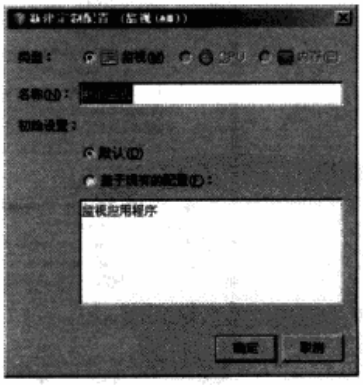


图 9-49 “新建定制配置”对话框

单击“确定”按钮，创建配置。在新配置上单击右键，打开弹出菜单，如图 9-50 所示。弹出菜单的几个命令的含义介绍如下。

- “重命名”命令：重命名配置，选择该命令后会打开“重命名配置”对话框，如图 9-51 所示。

Download at http://www.pin5i.com/

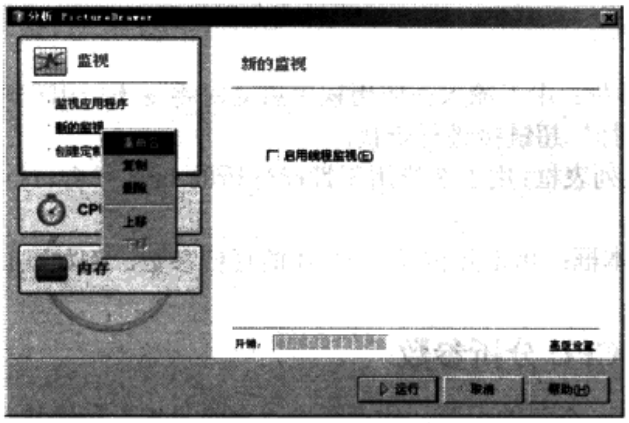


图 9-50 配置的弹出菜单

- “复制”命令：复制当前的配置，并另存为一个新的配置。选择该命令后会打开“重复的配置”对话框，如图 9-52 所示。
- “删除”命令：删除选中的配置。
- “上移”命令：把当前配置向上移动一个位置。



图 9-51 “重命名配置”对话框

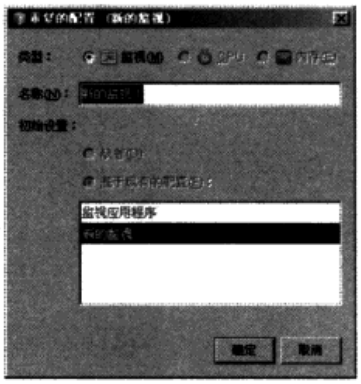


图 9-52 “重复的配置”对话框

9.6.2 自定义监视分析参数

在监视分析中创建一个新的配置，然后单击“高级设置”超链接，打开监视性能分析的自定义参数界面，如图 9-53 所示。

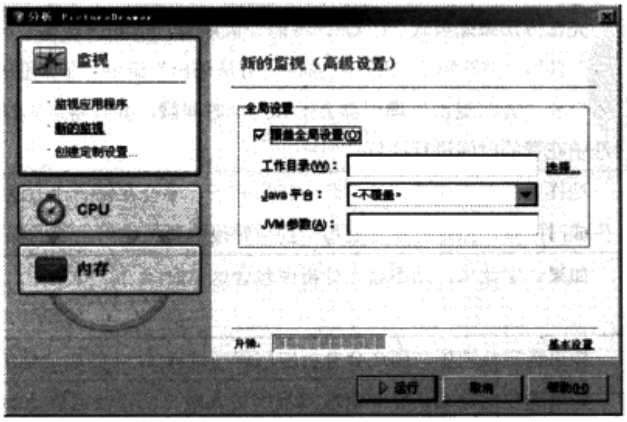


图 9-53 自定义监视分析参数界面

Download at http://www.pin5i.com/

选中“覆盖全局设置”复选框，则所做的修改会覆盖全局设置。对话框中的各个参数说明如下。

- “工作目录”文本框：用于输入在应用程序启动时将变为应用程序当前目录的工作目录。也可以单击“选择”超链接进行查找。
- “Java 平台”下拉列表框：用于选择用于性能分析的 Java 平台，选中的平台将取代 Profiler 默认的平台。
- “JVM 参数”文本框：指定要传递给 JVM 的任何参数，例如默认最大堆的大小。



9.6.3 自定义 CPU 分析参数

在 CPU 分析中创建一个新的配置，然后单击“高级设置”超链接，打开 CPU 性能分析的自定义参数界面，如图 9-54 所示。自定义 CPU 分析参数的说明如表 9-2 所示。

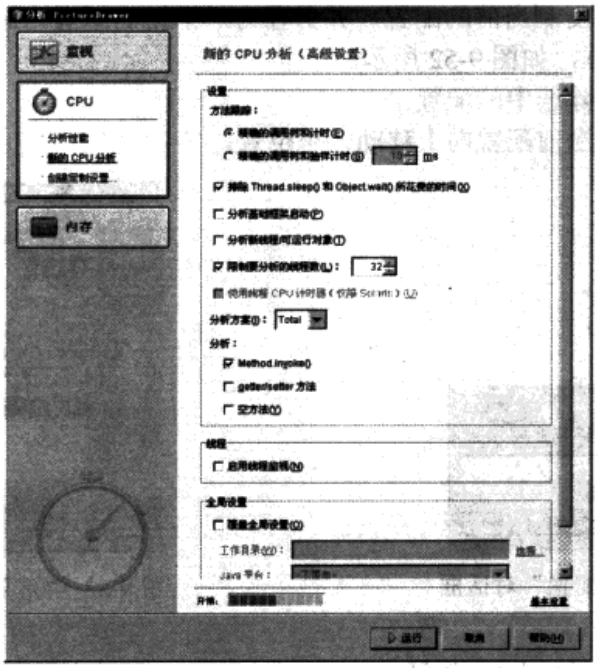


图 9-54 自定义 CPU 分析参数界面

表 9-2 自定义 CPU 分析参数的含义

参 数	含 义
精确调用树和计时	完整方法跟踪模式。在 CPU 性能分析期间，Profiler 在每个分析的方法开头插入“方法进入”调用，并在每个 return 前插入“方法退出”调用。在完整的分析模式下，每个“方法进入”和“方法退出”调用都会生成一个时间戳，并且将这些时间戳之间的差值作为在此方法中花费的时间进行计算
精确调用树和抽样计时	抽样方法跟踪模式。在这种分析模式下，不会在每个“方法进入”/“方法退出”调用上生成时间戳，而是在指定的每个抽样时段快要结束时出现的那些调用上生成时间戳
排除在 Thread.sleep()和 Object.wait()中花费的时间	如果选定此项，则不会在分析中包含这些时间
分析基础框架启动	如果选定此选项，则还会分析应用程序目标 JVM 的启动性能
分析新线程/可运行对象	如果选定此选项，将分析应用程序进入根分析方法后启动的任何线程或可运行对象的代码。在分析整个应用程序时，将默认选择此选项

Download at http://www.pin5i.com/

续表

参数	含义
限制分析的线程数	可以限制分析的线程数。每次在以前未注册的线程进入分析的代码时，此计数值都会减少。当它达到 0 时，将停止注册新线程。仅为注册的线程收集性能分析数据。在 NetBeans 主菜单中选择”分析”→“重置收集的结果”命令，会将计数器重置为此值
使用线程CPU计时器(仅限 Solaris)	可以使用不同种类的操作系统计时器来生成方法进入/退出事件的时间戳。默认情况下，使用绝对计时器来测量绝对时间或 Wall Clock 时间。线程 CPU 计时器只测量执行给定线程所花费的时间。实际上，每次 CPU 停止执行代表给定线程的代码时，操作系统都将暂停该计时器，并在每次执行切换回该线程时恢复执行。该选项只有在 Solaris 操作系统中才有效。对于 Windows 操作系统，该选项为禁止状态
Method.invoke()	如果选定此项，将跟踪并分析所有调用，即使用 Java 反射，java.lang.reflect.Method.invoke()
getter/setter 方法	如果选定此项，将分析包含 getXXX 或 setXXX 对象字段的方法
空方法	如果选定此项，将分析不包含任何可执行代码的方法
启用线程监视	监视线程，可以获取有关系统和应用程序线程活动的更多详细数据

“分析方案”下拉列表框用来选择分析方案，以限制分析的方法数。通常，最理想的情况是，分析的方法数与分析应用程序时实际调用的方法数相同或非常接近。可以选择以下分析方案。

- Lazy：默认的分析方案，最适于分析长时间运行的应用程序。使用此方案时，Profiler 将按照以下方式来分析方法。首先分析根方法。然后，在第一次执行任意分析的 x()方法时，先扫描其代码，然后依次分析 x()可能调用的所有方法。通过这种方法，可以确保分析的方法数近似于目标应用程序在其执行生命周期内实际调用的方法数。通过最大限度地减少已分析但未从根方法中调用的方法数，可以减少分析时间。
- Total：选定这种方案时，Profiler 与大多数其他分析器的工作方式类似，即在装入包含目标应用程序方法的类时，Profiler 将立即分析所有这些方法。此方案更适合分析应用程序启动时的性能情况或分析整个应用程序（如短时间运行的命令行实用程序），因为在分析长时间运行的应用程序时产生巨大的开销。
- Eager：Eager 分析是以上两种方案的折衷方案。与 Lazy 方案不同，装入某个类后，Eager 方案会立即在该类中查找所有可能执行的方法（即根方法可以直接或间接调用的方法）。通过 Eager 分析所有可能执行的方法，可能会使分析的方法比调用的方法多很多，因此建议不要使用此方案来分析长时间运行的应用程序。在应用程序启动期间执行的操作时，或者在 Total 分析产生过多的开销时，可以使用这种方案。



9.6.4 自定义内存分析参数

在内存分析中创建一个新的配置，然后单击“高级设置”超链接，打开内存性能分析的自定义参数界面，如图 9-55 所示。自定义内存分析参数的说明如表 9-3 所示。

表 9-3 自定义内存分析参数的含义

参数	含义
记录分配的栈跟踪	如果选定此项，在记录对象分配时，将记录调用栈。可以调整记录的调用栈的最大深度（即从顶部记录的框架数）来减少开销。该选项包含在“基本”设置中，需要在图 9-56 中单击“基本设置”超链接才能修改。只有选择了该选项后，才能够编辑“完整栈深度”参数和“将栈限制为 X 框架”参数
完整栈深度	深度不受限制

Download at http://www.pin5i.com/

续表

参　数	含　义
将栈限制为 X 框架	指定栈限制。例如通过将最大深度限制为较小的数字（如 3 到 5），可以大大减少性能分析开销
获取内存结果时运行垃圾回收	如果选定此项，每次在 NetBeans 主菜单中选择”分析”→“实时结果”命令时，都将在目标 Java 虚拟机中调用完整垃圾回收。此选项会有助于了解更实际的对象活动性情况，但缺点是它还会增加开销
启用线程监视	监视线程，可以获取有关系统和应用程序线程活动的更多详细数据

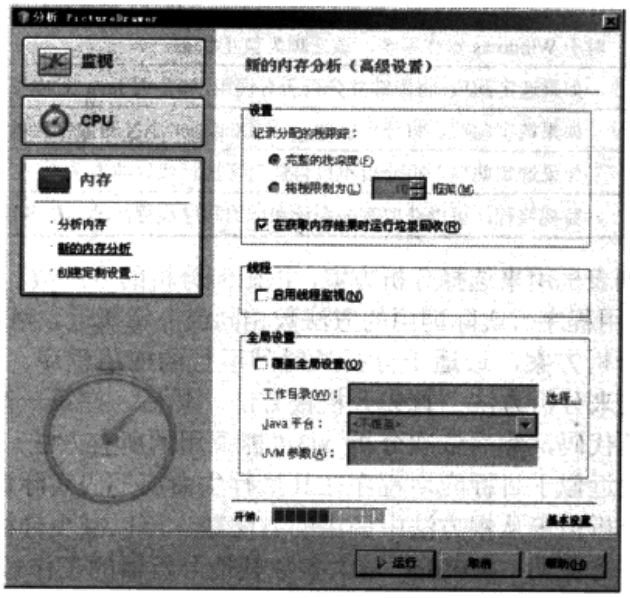


图 9-55　自定义内存分析参数

9.7　将 Profiler 连接到运行的应用程序

Profiler 还提供了连接模式，可以将该工具连接至在本地或远程计算机上运行或即将启动的应用程序。这个功能非常重要，因为可以使得应用程序不必重新启动就可以进行性能分析。这使得 Profiler 非常适于分析运行时间长的应用程序，尤其适于分析在应用服务器上运行的 Web 应用程序或企业应用程序以及应用服务器本身的代码。

9.7.1　使用连接向导进行直接连接

在 NetBeans 主菜单中选择“分析”→“连接 Profiler”命令，打开“选择性能分析任务”对话框，如图 9-56 所示。

单击“定义”超链接，打开“选择目标类型”对话框，如图 9-57 所示。

在“目标类型”下拉列表框中可以选择被分析的目标应用程序的类型，可以有如下几种类型。

- 应用程序：Java 标准应用程序。
- Applet：Java 小程序。
- J2EE Web/应用服务器：Java EE Web 应用程序，或者为应用服务器。

这里选择“应用程序”，会在窗口下面显示目标设置，如图 9-58 所示。

Download at http://www.pin5i.com/

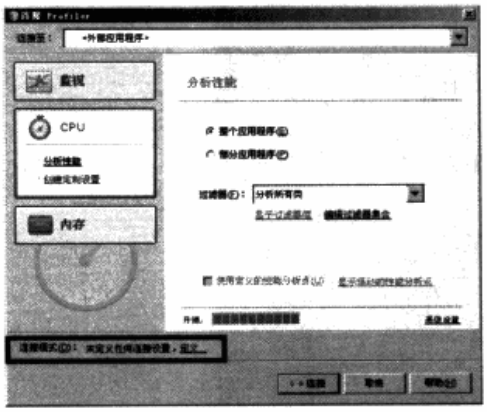


图 9-56 “选择性能分析任务”对话框

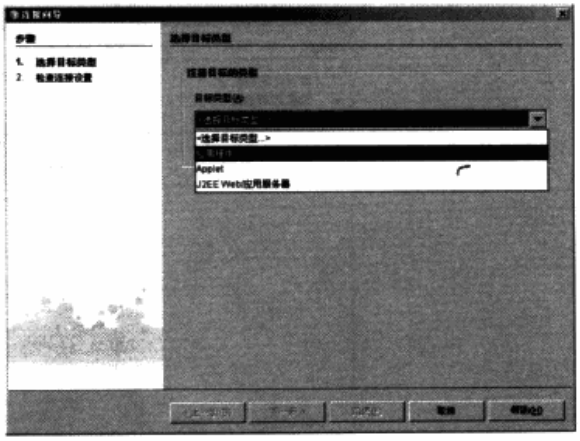


图 9-57 “选择目标类型”对话框

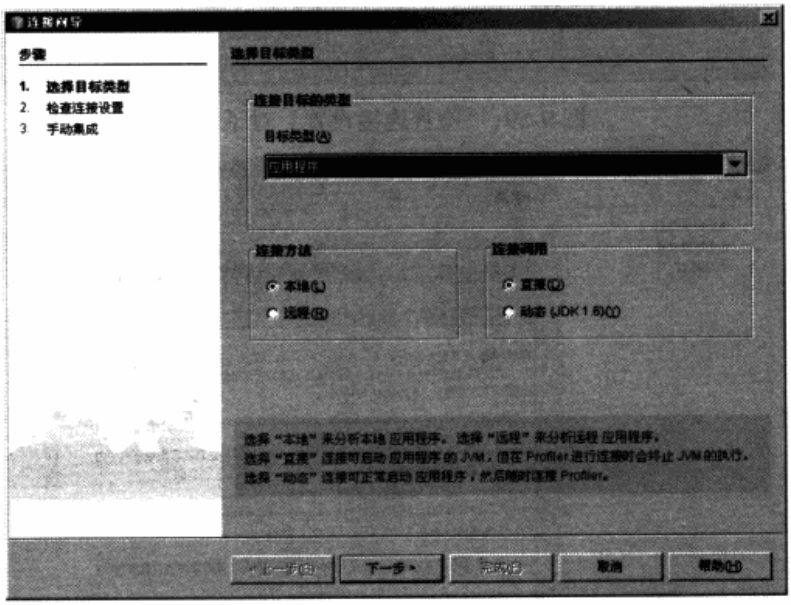


图 9-58 显示目标设置

如果要进行分析的程序位于本地，则要选中“本地”单选按钮；如果要进行分析的程序位于其他计算机，则要选中“远程”单选按钮。

如果有下面的几种情况之一，则必须选择“直接”单选按钮：

- 分析远程计算机上的应用程序；
- 分析使用 Java 虚拟机版本 5 或者更高版本的应用程序；
- 要将 Profiler 提前连接至该应用程序。例如，对应用程序的启动情况进行性能分析。

如果将 Profiler 连接至已运行的本地应用程序（Profiler 不会启动目标应用程序），则需要选中“动态（JDK 1.6）”单选按钮。

关于进行动态连接的具体方法将在下一小节进行介绍。这里选中“直接”单选按钮，单击“下一步”按钮，打开“检查连接设置”对话框，如图 9-59 所示。

在该对话框中，列出了一些连接参数。在检查无误后，单击“下一步”按钮打开“手工集成”对话框，如图 9-60 所示。

该对话框中给出了手工集成的详细操作步骤。单击“完成”按钮，关闭连接向导，返回“选择性能分析任务”对话框。

Download at http://www.pin5i.com/

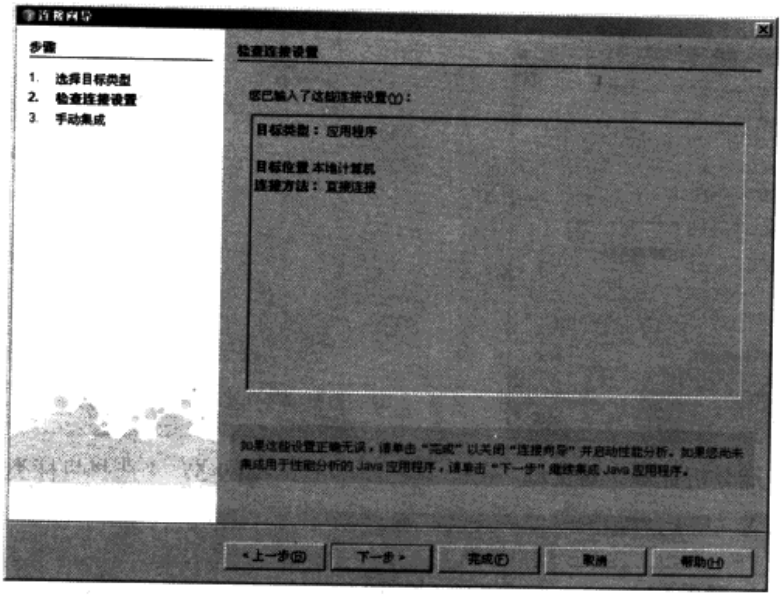


图 9-59 “检查连接设置”对话框

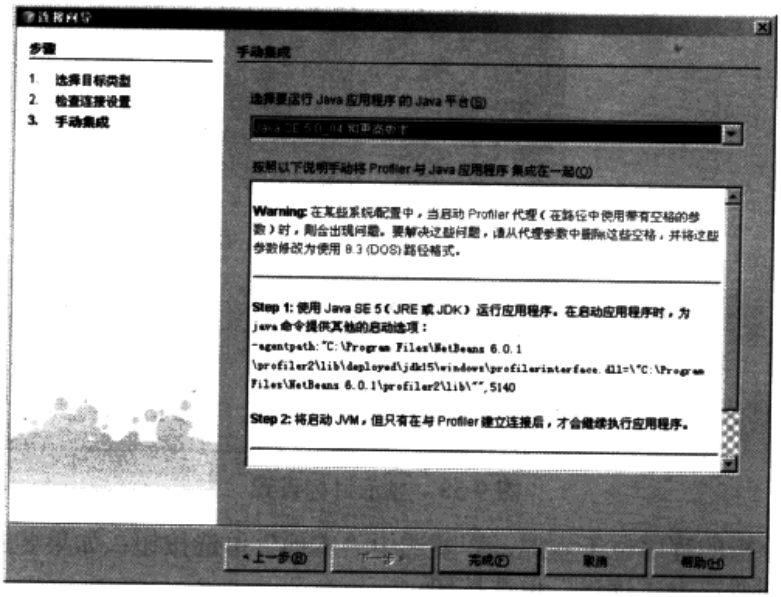


图 9-60 “手工集成”对话框

例如要执行的程序为“anagrams.jar”，那么复制图 9-61 中的命令，在项目目录下创建一个批处理文件 run.bat，内容如下所示：

```
    java -agentpath:"C:\Program Files\NetBeans 6.0\profiler2\lib\deployed\jdk15\
windows\profilerinterface.dll=\"C:\Program Files\NetBeans 6.0\profiler2\lib\"",
5140
    -jar "dist\anagrams.jar"
```

双击 run.bat 文件，显示运行界面如图 9-61 所示。此时，表示应用程序已经与 Profiler 集成在一起了。此时不会显示应用程序的执行界面，只有在启动了 Profiler 后应用程序才会继续执行。

在如图 9-56 所示的“选择性能分析任务”对话框中，单击“连接”按钮，启动 Profiler 开始进行性能分析。在分析结束后，显示如图 9-62 所示的对话框。单击“确定”按钮，即可完成此次性能分析。

Download at http://www.pin5i.com/



图 9-61 启动应用程序

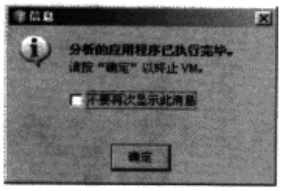


图 9-62 性能分析结束时的提示对话框

9.7.2 进行动态连接

上一节介绍的方法必须重新启动应用程序。对于一些应用程序（例如某个服务器程序），需要保持程序始终运行，不能中断。此时，可以使用动态连接的方法进行性能分析。在 NetBeans 6.0 中，要使用动态连接功能，需要 JDK 6.0 或者以上版本。在如图 9-58 所示的对话框中，选择“动态（JDK 1.6）”单选按钮，然后单击“下一步”按钮，完成向导并返回“选择性能分析任务”对话框。

单击“连接”命令，打开“选择进程”对话框，如图 9-63 所示。单击“刷新”命令，可以刷新进程列表。在下拉列表框中选择要被分析的应用程序，单击“确定”按钮，即可动态地连接到应用程序，进行性能分析了。

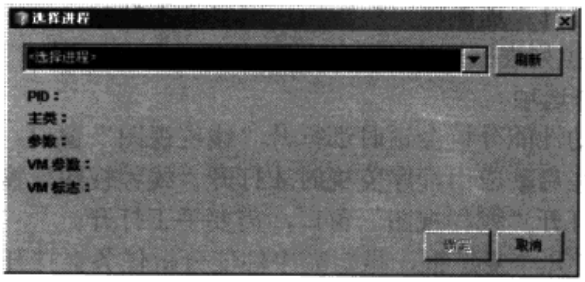


图 9-63 “选择进程”对话框

说明：在 NetBeans 5.x 中，如果要运行动态连接功能，需要先下载并安装专门为 Profiler 修改的 JVM。因此，使用起来比较麻烦。NetBeans 6.0 中已经改进了这个功能，开发人员可以直接使用动态连接功能，分析已经运行的应用程序。

9.8 配置 Profiler 选项

在 NetBeans 中可以配置 Profiler 的一些参数，例如指定 Profiler 所使用的 Java 平台、进行通信的通信端口、生成快照时的默认动作等。

通过从主菜单中选择“工具”→“选项”命令，打开“选项”对话框。在“选项”对话框的上部选择“杂项”按钮，再打开 Profiler 标签页，显示 Profiler 的设置，如图 9-64 所示。

“Profiler Java 平台”下拉列表框用于指定在分析应用程序时使用的目标 Java 虚拟机，默认为 NetBeans 集成开发环境使用的虚拟机。“通信端口”文本框用于指定 Profiler 所侦听的通信端口，默认端口为 5140。

Download at http://www.pin5i.com/

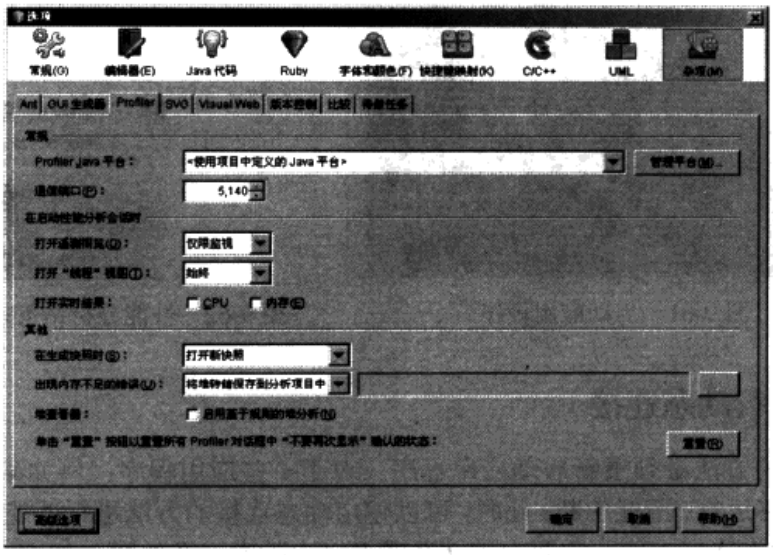


图 9-64 配置 Profiler 选项

“打开遥测概览”下拉列表框用来指定在性能分析会话启动时，“遥测概览”窗口的的默认打开情况。可以有如下几个选项。

- 始终：在每次启动性能分析会话时都打开“遥测概览”窗口。
- 仅限监视：只有在启动应用程序监视时才打开“遥测概览”窗口。
- 从不：从不自动打开“遥测概览”窗口，需要手工打开。

“打开线程视图”下拉列表框用来指定在性能分析会话启动时，“线程视图”窗口的默认打开情况。可以有如下几个选项。

- 始终：在每次启动性能分析会话时都打开“线程视图”窗口。
- 仅限监视：只有在启动应用程序监视时才打开“线程视图”窗口。
- 从不：从不自动打开“线程视图”窗口，需要手工打开。

选中下面的一个或者几个复选框，指定哪个性能分析任务会打开“实时结果”窗口。

- CPU：进行 CPU 性能分析时，打开“实时结果”窗口。
- 内存：进行内存分析时，打开“实时结果”窗口。

如果要指定在 Profiler 控制面板中单击“生成快照”按钮时出现的行为，可以在“在生成快照时”下拉列表框中选择如下几个值。

- 打开新快照：选中该选项，会在“性能分析结果”窗口中打开新快照。
- 保存新快照：选中该选项，会在“快照”列表中保存该快照，而不会在“性能分析结果”窗口中打开它。
- 打开并保存新快照：选中该单选按钮，会在“性能分析结果”窗口中打开新快照，并在“快照”列表中保存该快照。

“出现内存不足的错误”下拉列表框用于指定发生 OutofMemoryError 错误时的处理方法，可以有如下几个选项。

- 不执行任何操作：什么也不做。
- 将堆转储保存到分析项目中：把内存数据保存在被分析的项目中。
- 将堆转储保存到临时目录中：把内存数据保存在临时目录中。
- 将堆转储保存到：把内存数据保存到指定的目录，选择该选项后，需要在后面的文本框中指定一个具体的目录。

在如图 9-62 所示的对话框中，如果选中“不要再次显示此消息”复选框，然后单击“确

Download at http://www.pin5i.com/

定”按钮关闭对话框，那么这个对话框下次将不再出现。Profiler 中有很多这样类似的对话框，如果希望这些对话框能够再次出现，那么单击“重置”按钮即可。

9.9 小结

性能分析对于提高程序的性能、找到程序中潜在的内存泄露等问题都很有帮助。利用 NetBeans 提供的 Profiler 性能分析工具，可以实现大多数情况的性能分析任务。本章对于具有一定 Java 编程经验并想进一步提高的读者比较适用。对于初学者，建议只进行快速浏览，以后再认真学习。通过本章的学习，读者应该掌握以下基本内容：

- 几个性能分析任务的使用方法
- Profiler 控制面板的使用
- Profiler 中几个窗口的作用以及使用方法
- Profiler 的配置方法

Download at http://www.pin5i.com/

第 10 章

在 NetBeans 中进行版本控制

CVS（Concurrent Version System）和 Subversion 都是非常优秀的版本控制系统，它可以在软件开发过程中记录文件的变更细节以及存储软件开发过程中的各种版本。版本控制系统允许多名开发人员同时编辑同一文件而不会丢失数据。每名开发人员都拥有该文件的副本，最后将所有的修改合并到一起成为最终的结果。CVS 和 Subversion 都基于 C/S，可以运行在 UNIX、Linux 以及 Windows 平台之上。CVS（Subversion）的服务器端存有一个源代码库，源代码库中保存众多不同的项目。客户端通过和服务器端建立连接，将源代码下载到本地计算机上，开发人员就可以在本地计算机上进行添加、修改、删除等操作，最后提交到服务器。

NetBeans 内置对 CVS 和 Subversion 的支持，NetBeans 可以作为 CVS 系统的客户端从而对开发进行版本控制。本章以 CVS 为主讲解如何在 NetBeans 集成开发环境中进行版本控制，在本章的最后部分介绍如何在 NetBeans 中使用 Subversion 以及其他的版本控制系统。

本章主要内容

- CVS 的基本工作原理
- 搭建 CVS 服务器
- 在 NetBeans IDE 中签入和签出项目源
- 更新文件和提交更改
- 查看文件状态和更改记录
- 使用标记和分支
- 使用 Subversion 以及其他版本控制系统

本章亮点

- 搭建 CVS 服务器
- 比较文件更改
- 查看文件历史版本记录
- 使用标记和分支

10.1 CVS 基本原理

在通过 CVS 进行版本控制时，应该首先搭建 CVS 服务器。目前，有很多的 CVS 服务器，例如 CVSNT 等。本书将带领读者搭建 CVSNT 服务器，并使用 NetBeans 连接此服务器讲解 CVS 的使用。

CVS 的工作原理如图 10-1 所示。CVS 服务器用于保存项目源（包括源代码、项目文件、文档等任何需要进行版本控制的对象）。但是 CVS 服务器中的项目源不能直接被使用。要将其导出到客户端，在客户端建立的副本才能对其进行编辑修改。

在进行团队开发时，需要对代码进行共享。通常开发人员首先将自己的代码保存（也叫导入或者签入，英文为 Check in）到 CVS 服务器，这样团队中的其他成员可以把这些代码导出（也

Download at http://www.pin5i.com/

叫签出，英文为 Check out）到本地计算机上。这样就可以在多个客户端上保存同一项目的副本，实现了代码的共享。

需要说明的是，CVS 服务器端和客户端可以在同一台机器上，也可以是在局域网中的不同机器上，或是在互联网中的不同机器上。

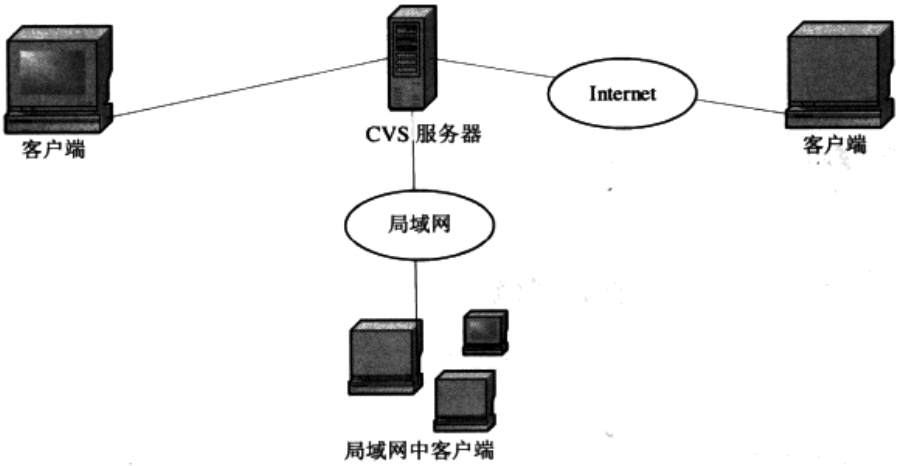


图 10-1 CVS 基本原理示意图

客户端可以使用多种方法与服务器建立连接。在建立连接时，需要给出服务器的地址，就像用 IE 浏览器访问 Web 网站时要输入网站的地址一样。CVS 服务器连接地址的格式定义如下所示：

```
[:访问方法:][[[用户名][:密码]@]主机名[:[端口]]]/路径
```

其中，连接地址各个部分的说明如下。

- 访问方法：类似于连接协议，NetBeans 支持以下访问方法。
 - local：连接到本地服务器。
 - ext：使用系统外部的方法（如 rsh）连接。
 - pserver：使用 pserver 方法连接。
 - fork：将本地机器作为远程机器来连接（通常用于测试）。
- 用户名：在 CVS 服务器上注册的用户名。
- 密码：注册用户时使用的密码。
- 主机名：CVS 服务器的主机名称或 IP 地址。
- 端口：连接服务器时使用的端口号。如果为空表示使用默认端口。
- 路径：在 CVS 服务器中注册的用于存放项目的仓库路径。

例如，要使用用户名 wind9494 访问路径为 cvs.dev.java.net/cvs 的仓库，那么连接地址应该为：pserver:wind9494@cvs.dev.java.net:/cvs。

10.2 搭建 CVS 服务器

下面以 CVSNT 为例搭建 CVS 服务器，CVSNT 可以在 http://www.cvsnt.com 下载，笔者使用的版本为 cvsnt-2.0.58d，在本地计算机上搭建，然后使用本地计算机上的 NetBeans 集成开发环境连接此服务器。

1. 双击安装程序，安装 CVSNT，按照安装向导的提示逐步完成安装即可。
2. 从开始菜单启动“Service control panel”，如图 10-2 所示。

Download at http://www.pin5i.com/

❸ 单击“Service Status”标签，确保服务已经启动，如图 10-3 所示。

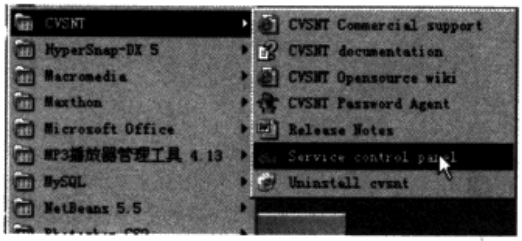


图 10-2 启动“Service control panel”

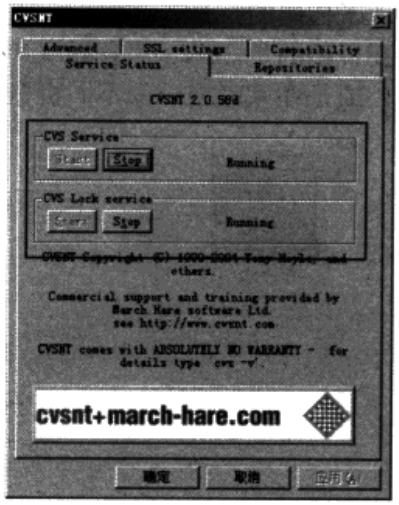


图 10-3 查看服务器状态

❹ 单击“Repositories”标签，然后单击“Add”按钮，打开“Edit repository”对话框，如图 10-4 所示。在“Location”文本框中输入项目仓库在服务器上的存放路径。在“Name”文本框中输入此仓库映射的名称。单击“OK”按钮返回，这样就添加了一个项目仓库，如图 10-5 所示。

❺ 其他配置保持默认，单击“确定”按钮完成对 CVSNT 的配置，至此 CVS 服务器搭建完毕。

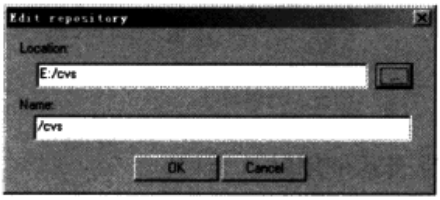


图 10-4 “Edit repository”对话框

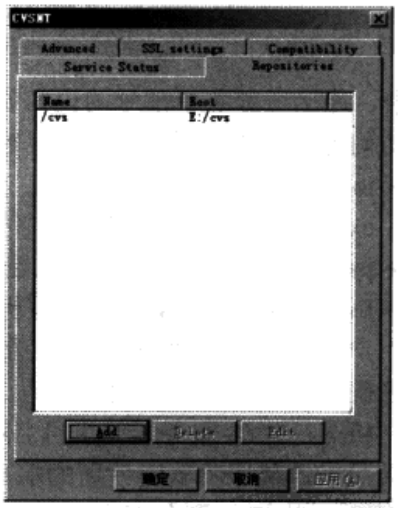


图 10-5 添加一个项目仓库

10.3 签入和签出源

签入项目是将本地机器上的项目导入到 CVS 服务器的库中，与其他开发人员共享项目源代码。与签入相反，签出是将 CVS 服务器库中的项目导出到本地机器上，实现多人同时协同开发。

10.3.1 签入

启动 NetBeans 集成开发环境，新建一个项目，此处新建的项目是“样例”中的“Java”项目“GUI 窗体示例”，命名为“GUIDemo”。

Download at http://www.pin5i.com/

在“项目”窗口中右击“GUIDemo”项目节点，选择弹出菜单“版本控制”→“导入到 CVS 资源库”命令，打开“导入项目选项”对话框。在“CVS 根”文本框中输入连接路径，这里连接到第 10.2 节中在本机上搭建的 CVSNT 服务器。也可以单击右侧的“编辑”按钮在打开的“编辑 CVS 根”对话框中编辑 CVS 根，如图 10-6 所示。在“访问方法”下拉列表中选择访问方法，这里选择“pserver”。在“用户”文本框中输入注册的用户名，这里使用操作系统用户名。在“主机”文本框中输入服务器地址，因为服务器在本机上，所以此处使用 localhost 或 127.0.0.1。在“端口”文本框中输入端口号，如果为空，表示使用默认端口。在“资源库路径”文本框中输入到连接到的库的路径，此处使用在服务器上注册的“/cvs”。单击“确定”按钮返回“导入项目选项”对话框。

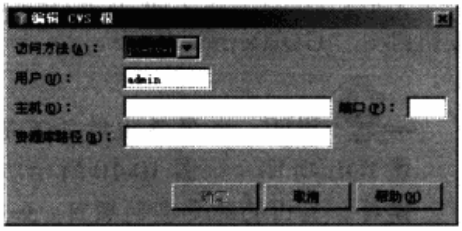


图 10-6 “编辑 CVS 根”对话框

在“口令”文本框中输入操作系统用户的密码。单击“下一步”按钮，弹出“要导入的文件夹”对话框。在“要导入的文件夹”文本框中输入要导入到 CVS 库的本地文件夹。在“导入消息”文本框中输入必要的消息。在“库文件夹”文本框中输入要将本地文件夹导入到库中的哪个文件夹中，如果输入的文件夹不存在，则会在服务器库中新建此文件夹。取消选择“导入后执行签出”复选框，关于如何签出将在后面进行讲解。

单击“完成”按钮，开始签入任务，这时会在“输出”窗口中显示签入的信息，如图 10-7 所示。

在服务器上打开“我的电脑”，进入到“E:\cvs”也就是项目仓库的存放位置，会发现项目库的文件夹里多了一个刚刚签入的项目文件夹，这证明项目“GUIDemo”已经被签入到 CVS 库中，如图 10-8 所示。

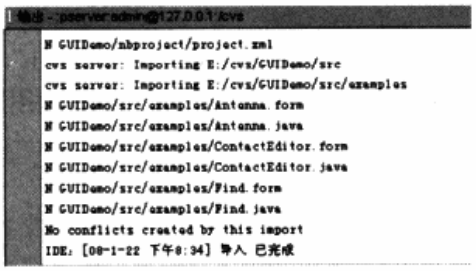


图 10-7 签入的输出信息

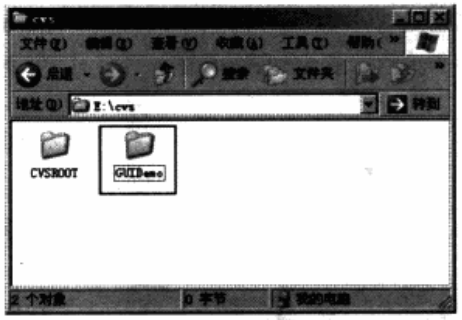


图 10-8 签入的项目文件夹

10.3.2 签出

在“项目”窗口中，关闭“GUIDemo”项目，下面使用签出命令将刚才签入的“GUIDemo”项目签出。

❶ 从主菜单中选择“版本控制”→“CVS”→“签出”命令，打开“CVS 根”对话框。这里的配置和“签入”时一样，单击“下一步”按钮。

❷ 在“要签出的模块”对话框中的“模块”文本框中输入要签出的模块，模块指的就是 CVS 库中的一个个项目，空表示所有模块，也可以单击右侧的“编辑”按钮在打开的“浏览 CVS 模块”对话框中选择。这里输入刚刚签入的“GUIDemo”。在“分支”文本框中输入项目分支（分支将在本章后面的部分具体讲解），也可以通过单击右侧的“浏览”按钮来选择。在“本地文件夹”文本框中输入本地 CVS 的工作目录，也就是模块签出到本地文件夹的位置。单

Download at http://www.pin5i.com/

击“完成”按钮开始签出任务。

❸ 在“输出”窗口同样会显示签出信息。如果签出的是一个 NetBeans 项目，那么模块签出完毕后，NetBeans 会提示是否将它作为项目打开，如图 10-9 所示，此处选择“打开项目”按钮打开“GUIDemo”项目。



说明：如果签出的并不是 NetBeans 项目，NetBeans 会询问是否使用签出的源来创建 IDE 项目，如图 10-10 所示。单击“创建项目”按钮，使用打开的“新建项目”向导创建符合签出的源类型的项目。如果不想在签出完成之后就打开项目，也可以在以后通过打开项目命令打开签出的模块。NetBeans 会自动识别项目是否使用了版本控制。

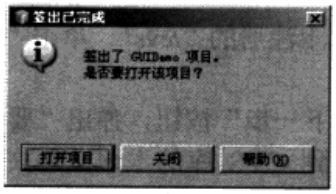


图 10-9 打开签出的项目

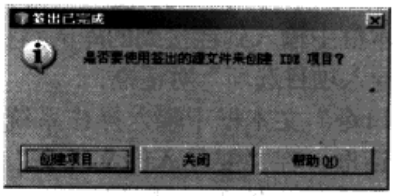


图 10-10 使用签出的源创建项目



10.3.3 通过 CVS 浏览 NetBeans 源代码

netbeans.org 上的 CVS 服务器保存了整个 NetBeans 的源代码，通过 NetBeans 集成开发环境可以将其签出到本地计算机上，其库连接路径为“:pserver:用户名@cvs.netbeans.org:/cvs”。开发人员可以匿名访问这些项目，也可以在 netbeans.org 上注册用户，使用注册的用户名和密码，用户只有读的权限，如果想和其他开发人员一起开发 netbeans.org 上的项目，必须获得相应的权限。下面签出一个 NetBeans 项目。



图 10-11 “浏览 CVS 模块”对话框

❶ 选择主菜单“版本控制”→“CVS”→“签出”，在“CVS 根”对话框中输入“CVS 根”和口令，如果使用匿名用户访问，用户名为 anoncvs，口令为空，单击“下一步”按钮。

❷ 在“要签出的模块”对话框中单击“模块”文本框右侧的“浏览”按钮，在打开的“CVS 模块”对话框中可以看到如“profiler”、“visualweb”、“ide”等所有模块，如图 10-11 所示。

❸ 可以使用同样的方法选择“分支”。选择完成后单击“完成”按钮开始签出。如果项目比较大，可能需要很长时间，请耐心等待。

❹ 签出完成打开项目或从签出的源创建新项目，现在可以在 NetBeans 集成开发环境中查看、编辑 NetBeans 的源代码了。



10.3.4 查看文件状态信息

在 NetBeans 集成开发环境中可以方便地查看文件在 CVS 版本控制下的状态信息。当新建文件、修改文件或删除文件以及做出其他修改后，文件会以不同的颜色和标记显示出来，可以在“项目”、“文件夹”、“收藏夹”以及“CVS”等窗口中查看这些信息。

打开“GUIDemo”，在源包“examples”下新建一个“Java 类”，类名为“NewClass”。在

Download at http://www.pin5i.com/

“项目”窗口中会看到“NewClass.java”文件名显示为绿色，包节点和项目节点带有蓝色标记，选择 NetBeans 主菜单中的“视图”→“显示版本控制标签”命令，可以在文件名后显示文件的状态，如图 10-12 所示。

在源代码编辑器中打开“Find.java”，修改部分代码并保存，“Find.java”文件名会变为蓝色，状态标签显示为“修改”，如图 10-13 所示。

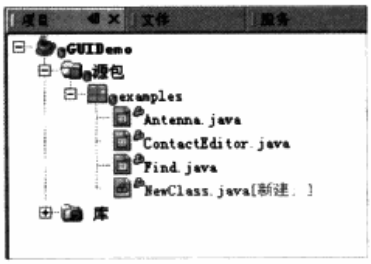


图 10-12 新建文件后“项目”窗口中显示的文件状态



图 10-13 修改文件后“项目”窗口中显示的文件状态

文件在不同的情况下会有不同的状态信息，这样方便开发人员查看文件的修改情况，表 10-1 对文件的不同颜色编码信息和标记进行了总结，供读者参考。

表 10-1 文件状态信息的颜色编码和标记

颜色编码和标记	描 述	含 义
HelpCtxKeys.java	黑色	文件正常，没有被修改过，处于更新状态
NewClass.java	绿色	新建的本地文件，在没有执行提交前不会存在库中
BiChildren.java	蓝色	已在本地更改的文件
NewClass.java	灰色	被忽略的文件，将不会对该文件执行版本控制命令（更新、提交等）
~~NewClass.java~~	删除线	提交时排除此文件，该样式只会在“版本控制”窗口和“提交”对话框中显示
Find.java	红色	包含冲突的文件（标记有<<<<<<和>>>>>>的部分）
GUIDemo	包或文件夹节点蓝色标记	包含新建文件或修改文件的包或文件夹。对于包，此标记只应用于包本身，而不应用于它的子包。对于文件夹，此标记表示该文件夹及其任何子文件夹中进行的本地修改
GUIDemo	包或文件夹节点红色标记	包含冲突文件的包或文件夹。对于包，此标记只应用于包本身，而不应用于它的子包。对于文件夹，此标记表示该文件夹及其任何子文件夹中进行的本地修改

10.4 显示、比较以及还原更改

开发人员最频繁的工作莫过于对文件的更改，在 NetBeans IDE 中，可以迅速显示文件被更改的信息，将更改的文件与以前的版本作比较，还可以还原不必要的更改。

10.4.1 显示更改

开发人员需要知道哪些文件以及文件的哪些部分被修改过了，将这些更改显示出来，在 NetBeans 集成开发环境中做如下操作：在“项目”窗口中的“GUIDemo”项目节点上单击鼠标右键，选择弹出菜单中的“CVS”→“显示更改”命令，此时会打开“CVS”窗口，并在其中显示文件更改的状态信息，如图 10-14 所示。

说明：可以选择主菜单中的“窗口”→“版本控制”→“CVS”，打开“CVS”窗口，此时“CVS”窗口中会显示所有使用 CVS 版本控制的项目所做的更改。

Download at http://www.pin5i.com/

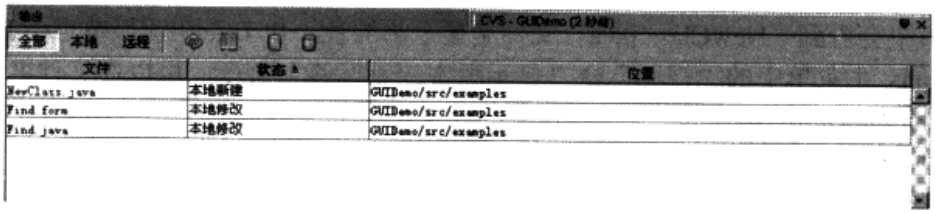


图 10-14 “版本控制”窗口

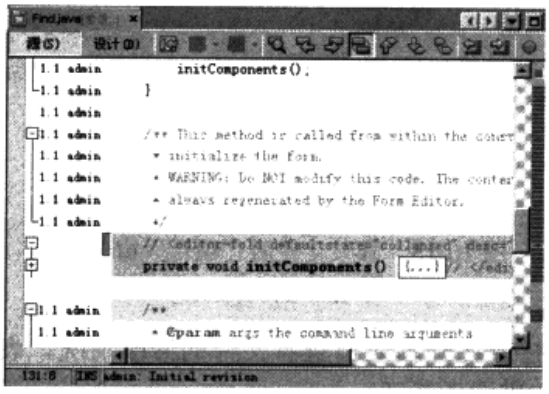


图 10-15 显示标注

在“CVS”窗口中，选择“全部”会显示本地以及远程服务器库中的更改，选择“本地”则只显示在本地计算机上所做的更改，选择“远程”只显示在远程服务器库中所做的更改。在被本地文件更改并保存时，版本控制窗口中会自动刷新显示哪些文件做了修改，单击“刷新状态”按钮会刷新在本地和远程服务器库中文件的状态。

在“CVS”窗口中右键单击文件，选择弹出菜单中的“显示标注”命令，会在源代码编辑器的左侧显示文件的版本号和修改者的名字，如图 10-15 所示。

10.4.2 比较更改

在 NetBeans 集成开发环境中可以轻松比较文件与以前版本相比在哪些地方做了改动，在“CVS”窗口中双击文件或单击“全部比较”按钮 比较所有文件。也可以在“项目”、“文件”、“收藏夹”窗口，在要比较的文件节点上单击鼠标右键，选择弹出菜单中的“CVS”→“比较”命令。

这时编辑窗口会被分为两个小窗口，比较文件修改过的部分，可以在下拉列表框中选择要比较的文件，使用“上一个差异”按钮 和“下一个差异”按钮 逐步查看所有更改，如图 10-16 所示。

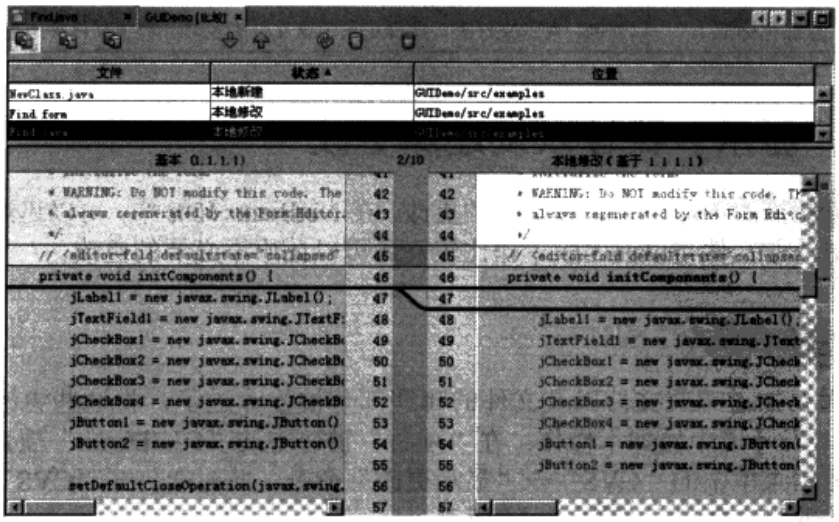


图 10-16 比较更改

修改过的行会使用不同颜色的背景显示，“蓝色”表示更改的行；“绿色”表示添加的行；“红色”表示删除的行。

Download at http://www.pin5i.com/

10.4.3 还原更改

还原更改是指将文件还原到修改前的状态，如果对文件进行了错误的修改，可以使用这条命令。

要还原新建的文件，只要删除该文件即可。要还原被删除的文件，在“版本控制”窗口中右键单击删除的文件，选择“还原删除”命令。对已存在的文件，在“项目”、“文件夹”、“收藏夹”窗口中，右键单击要还原的文件节点（非新建和删除的文件），选择弹出菜单中的“CVS”→“还原修改”命令，会弹出如图 10-17 所示的对话框。

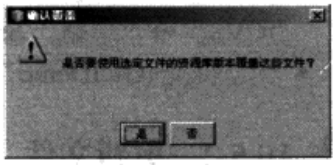


图 10-17 还原操作弹出的对话框

单击“是”按钮就会还原到修改前的状态，这其实等同于执行了一次更新操作，使用库中的文件覆盖修改过的文件。

10.5 提交和更新文件

当开发人员在本地修改完文件之后，就需要把文件的修改应用到服务器的库中，使用提交命令就是把修改过的文件提交到远程库中。与提交命令方向相反的更新则是将远程库中最新的文件副本应用于本地文件，使本地文件保持最新的状态。

10.5.1 提交文件

在“项目”窗口中右键单击“GUIDemo”项目节点，选择弹出菜单中的“CVS”→“提交”命令。弹出“提交-GUIDemo”对话框，如图 10-18 所示。

在这个对话框中会列出所有新建的文件、修改的文件和在本地删除的文件。可以对新建的文件执行“作为文本添加”、“作为二进制内容添加”和“从提交中排除”的提交操作。也可以对修改的文件执行“提交”和“从提交中排除”的操作。

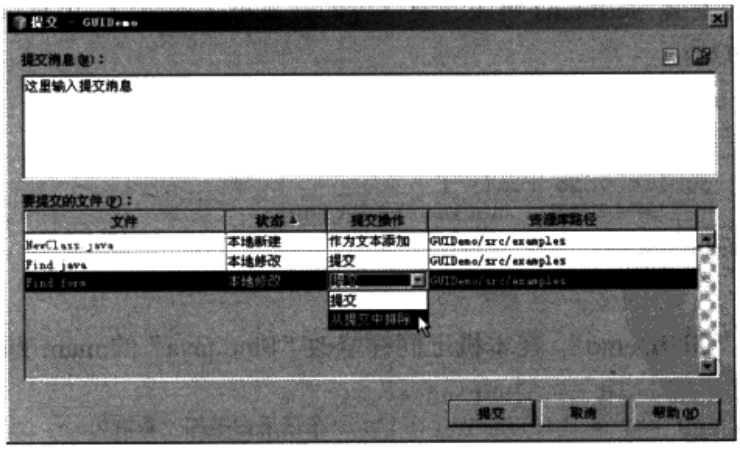


图 10-18 “提交-GUIDemo”对话框

完成必要的操作之后，单击“提交”按钮，开始提交任务。提交完成后“版本控制”窗口中的文件会变为空，同时在“输出”窗口中也会显示提交的信息。

可以在“CVS”窗口中单击“全部提交”按钮，提交在“CVS”窗口中所有列出的文件。

要提交特定文件，在“项目”、“文件夹”或“收藏夹”窗口中右键单击要提交的文件节点，

Download at http://www.pin5i.com/

选择弹出菜单中的“CVS”→“提交”命令，可以使用 Ctrl 或 Shift 键选择提交多个文件。

如果不想让文件使用版本控制，也就是不受 CVS 命令的影响，可以忽略这个文件。可以通过下面几种方式忽略文件。

- 在“项目”、“文件”和“收藏夹”窗口中，右键单击要忽略的文件，选择弹出菜单中的“CVS”→“忽略”命令，此时文件的颜色会变为灰色。
- 在“提交-GUIDemo”对话框中，将文件的“提交”操作改为“从提交中排除”。

10.5.2 更新文件

更新文件也许是开发人员每天要做的第一件事情。可以通过下面几种方式更新文件。

- 更新所有打开的项目。可以在“CVS”窗口中单击“全部更新”按钮 更新在“CVS”窗口中的所有文件。
- 更新特定项目。在“项目”窗口中右键单击要更新的项目节点，选择弹出菜单中的“CVS”→“更新”命令，或者在 NetBeans 主菜单中选择“版本控制”→“更新”命令。
- 更新特定文件。在要更新的文件或文件夹上单击鼠标右键，选择弹出菜单中的“CVS”→“更新”命令。可以使用 Ctrl 或 Shift 键选择更新多个文件。



说明：在实际开发中一定避免长时间不进行提交和更新操作，每天工作之前一定要执行更新操作。时常提交和更新可以降低开发中发生冲突的可能性。

10.5.3 解决冲突

无论是提交和更新，还是将要在本章后面部分讲到的合并分支，其实都是以行为单位进行合并。如果更改之处位于不同的行，此系统会依据实际情况对各行进行添加、修改、删除等操作。如果在服务器仓库中的文件副本与本地计算机中的文件副本相应的行有不同的内容，文件合并时就会发生冲突。

发生冲突有时是不可避免的，但也有办法解决它们。下面笔者将演示冲突的发生以及如何解决冲突。在另一台安装 NetBeans 集成开发环境的计算机上签出“GUIDemo”，修改“Find.java”的 main 方法如下所示：

```
public static void main(String args[]) {
        String str = "Do it yourself!";//在这里添加一条语句
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Find().setVisible(true);
            }
        });
    }
```

修改之后提交“GUIDemo”。在本机上同样修改“Find.java”的 main 方法，内容如下所示：

```
public static void main(String args[]) {
        String str = "Hello World!";//在这里也添加一条语句
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Find().setVisible(true);
            }
        });
    }
```

修改完成后，更新“GUIDemo”。这时会弹出警告提示框，如图 10-19 所示。

Download at http://www.pin5i.com/

此时在“CVS”窗口中“Find.java”的文件名变为红色，状态变为“本地冲突”。在源代码编辑器中打开“Find.java”，冲突的代码会显示在<<<<<<<和>>>>>>>之间。

```
    public static void main(String args[]) {
<<<<<<< Find.java
        String str = "Hello World!";
=======
        String str = "Do it yourself!";
>>>>>>> 1.11
        java.awt.EventQueue.invokeLater(new
Runnable() {
            public void run() {
                new Find().setVisible(true);
            }
        });
    }
```

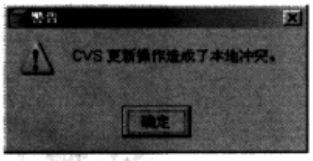


图 10-19 更新操作造成的冲突

在 NetBeans 集成开发环境中解决冲突非常简单，在“CVS”窗口中，右键单击“Find.java”文件，选择弹出菜单中的“解决冲突”命令，会打开“合并冲突解决程序”窗口，如图 10-20 所示。

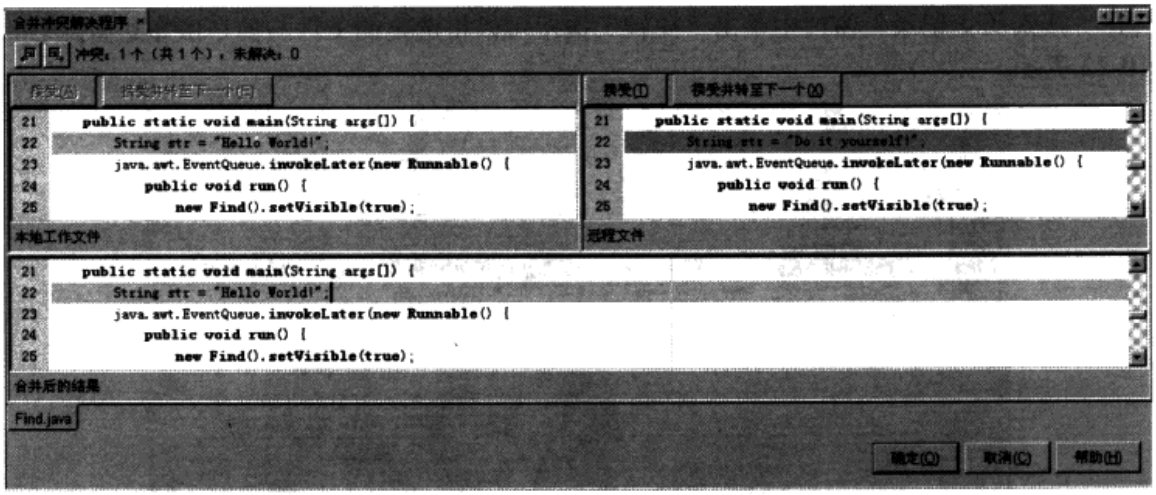


图 10-20 “合并冲突解决程序”窗口

如果要保留本地文件的修改，单击“本地工作文件”上面的“接受”按钮，同样也可以单击“远程文件”上的“接受”按钮接受远程文件的修改。在下方的窗口中会预览执行操作后文件合并的结果。如果有多个冲突可以单击“上一个差异”按钮 和“下一个差异”按钮 在各个冲突之间跳转。做完所有“接受”操作之后，单击“确定”按钮，在弹出的确认提示框中选择“是”即可解决冲突。

10.6 查看文件历史版本记录

在开发过程中有时需要查看某个文件过去的版本记录或取回过去的某个修订版本。在“项目”窗口中右键单击“GUIDemo”，选择弹出菜单中的“CVS”→“搜索历史记录”命令，会打开“搜索历史记录”窗口，如图 10-21 所示。

可以在其中输入一些搜索条件，在“消息”文本框中输入文件提交时的消息，在“用户名”文本框中输入提交文件的用户名，在“从”文本框中输入执行提交的起始标记（关于“标记”会在本章的后面部分讲解）、分支、修订版本号或日期，在“到”文本框中输入执行提交的截止标记、分支、修订版本号或日期。如果不输入搜索条件，则搜索以前所有的修订版本。

Download at http://www.pin5i.com/

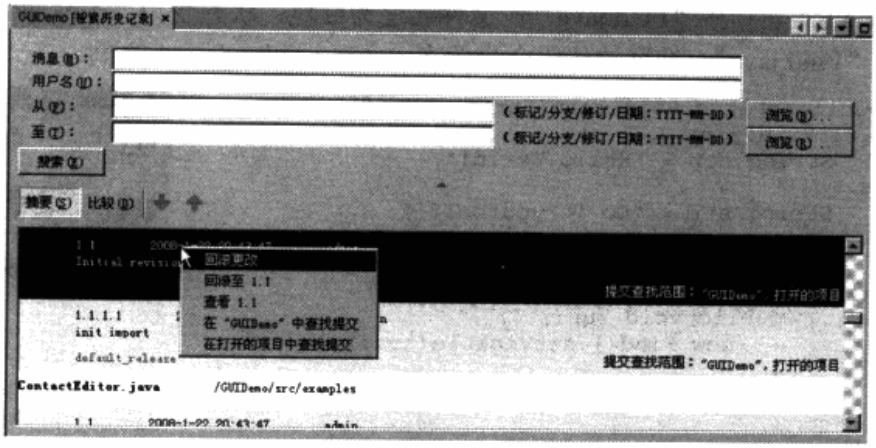


图 10-21 “搜索历史记录”窗口

在某个以前修订版本的文件上单击鼠标右键，可以执行弹出菜单的“回滚”等操作，例如，“回滚修改”会回滚到该版本修改前的状态，“回流至 x.x”可以将当前文件回滚到 x.x 版本。

在“搜索历史记录”窗口中单击“比较”按钮可以将选定的文件与前一版本进行比较，如图 10-22 所示。

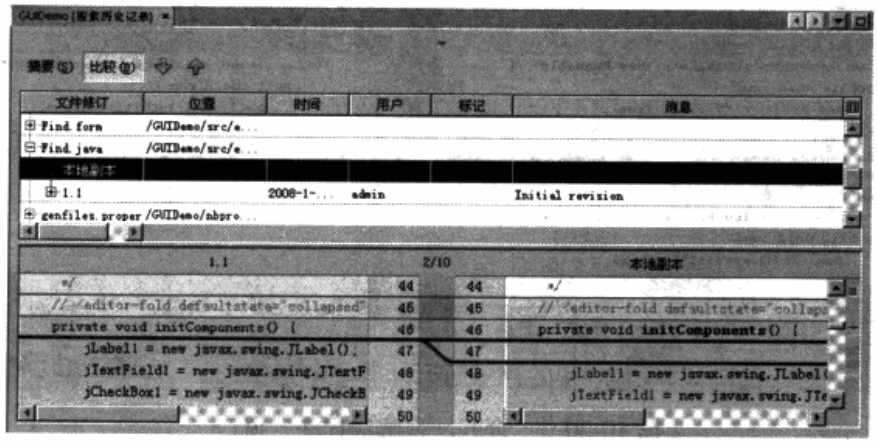


图 10-22 在“搜索历史记录”窗口中比较文件

10.7 使用标记和分支

“标记”是一个非常实用的版本控制功能，可以为特定修订版本的文件附加标记，以便日后开发查找，使用此功能还可以对某些版本的文件进行分组，将不同修订版本的文件附加上相同的标记从而分到同一组，实现分组开发。“分支”（branch）是指在对项目进行分组开发时，从项目中分出来的子开发线，而主开发线被称为“主干”（trunk），分支和主干的开发互不影响，分支开发和主干开发可以合并到一起。一个分支还可以再进行分支，主干和分支的关系如图 10-23 所示。

10.7.1 使用标记

在“项目”、“文件”和“收藏夹”窗口中，右键单击文件或文件夹节点，选择弹出菜单中的“CVS”→“标记”命令。在“标记名称”文本框中输入标记名称，也可以单击“浏览”按钮，从现有标记的列表中进行选择。如果输入的标记不存在，则会创建一个新标记，如图 10-24 所示。

Download at http://www.pin5i.com/

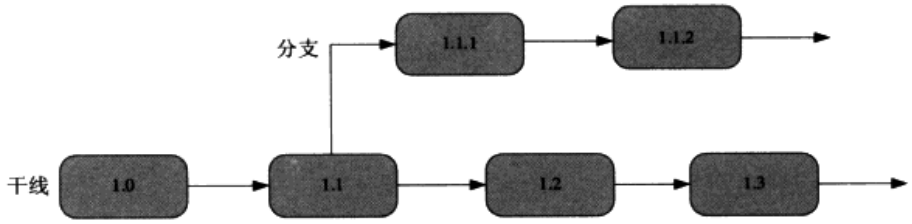


图 10-23 主干和分支的关系

如果要确保只将标记附加到最新的本地修订上，请选中“避免标记本地修改的文件”复选框。如果要将现有标记移到本地工作目录中的当前修订中，请选中“移动现有标记”复选框。

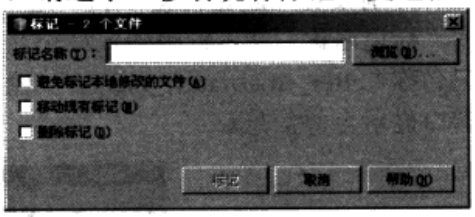


图 10-24 “标记”对话框

10.7.2 使用分支

要创建分支，在“项目”、“文件”或“收藏夹”窗口中，右键单击要作为分支根的文件或文件夹节点，然后选择弹出菜单中的“CVS”→“分支”命令，如图 10-25 所示。

在“分支”对话框中，输入分支名称或单击“浏览”按钮从可用分支列表中选择。此处输入的分支名称为“firstbranch”。

通常在创建分支的时候对其进行标记，方便日后使用，所以选中“创建分支前标记”复选框，然后在“标记名称”文本框中输入标记名称。如果要在创建新的分支后立即切换到该分支，请选中“随后切换到此分支”复选框。单击“分支”按钮，创建分支。

说明：*在添加分支时，远程库中会自动创建分支，而无须再执行“提交”操作。*

在开发过程中，可以在主干和各个分支之间进行切换。要切换分支，在“项目”、“文件”、“收藏夹”窗口中，右键单击文件或文件夹节点，选择弹出菜单中的“CVS”→“切换到分支”命令，如图 10-26 所示。



图 10-25 “分支”对话框



图 10-26 “切换到分支”对话框

在“切换到分支”对话框中，输入分支名称或单击“浏览”按钮从可用分支的列表中选择，单击“切换”按钮。若选择“切换到主干”单选按钮，则可以从分支切换到主干。

在开发过程中可以根据实际情况的需要将分支合并，可以将分支合并到主干中，也可以将主干合并到分支中，还可以将两个分支合并。在合并的过程中，一方的内容被融合到另一方中而自己的内容保持不变，例如，要将分支合并到主干中，分支的内容保持不变，而主干中的内容会包含分支中所做的更改。当进行合并操作时，强烈建议为分支附加上标记，这样方便日后查看。

为了演示分支的合并，按照上面的步骤再创建一个名为“secondbranch”的分支，然后切换到分支“firstbranch”，修改文件“NewClass.java”的内容如下所示：

Download at http://www.pin5i.com/


```
public class NewClass {

    /** Creates a new instance of NewClass */
    public NewClass() {
        System.out.println("This is a test!");//在这里添加一条语句
    }

}
```

在“项目”窗口中右键单击“GUIDemo”项目节点，选择弹出菜单中的“CVS”→“合并分支中的更改”，打开“合并分支中的更改”对话框，如图 10-27 所示。

“将更改合并到工作分支”文本框中显示要合并更改的分支，也就是当前正在编辑的分支。然后分别设置“起始位置”和“结束位置”。如果选中“标记合并后的主干”复选框并在其后的“标记名称”文本框中输入标记名称，则在合并工作完成之后为主干附加标记，此处输入标记名称“after_firstbranch_merge”。输入完成之后单击“合并”按钮开始合并，此时会在“输出”窗口显示合并信息。

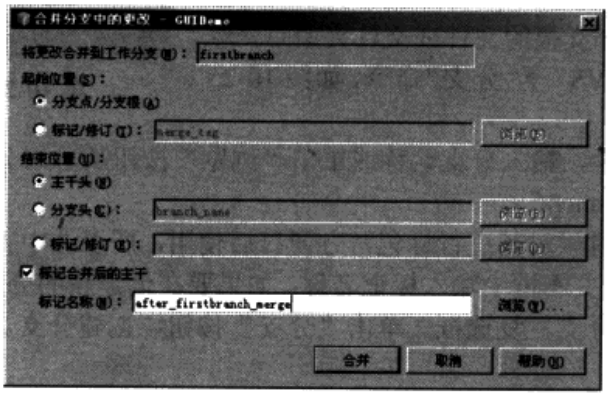


图 10-27 “合并分支中的更改”对话框

合并完成之后切换到主干，在源代码编辑器中打开“NewClass.java”，这时会发现代码内容和分支“firstbranch”中“NewClass.java”的内容一样，这说明分支“firstbranch”已经成功合并到主干中了。

用类似的方法可以实现分支与分支的合并，首先切换到“firstbranch”分支，按照上面的讲解的步骤打开“合并分支中的更改”对话框中，选择“从分支合并”单选按钮，并在其后的方框中输入分支名称“secondbranch”，单击“合并”按钮，即可将分支“firstbranch”中的内容合并到分支“secondbranch”中，这样就实现了两个分支的合并。

10.8 使用 Subversion

Subversion 的使用方法和 CVS 相似，本章不对 Subversion 的使用进行详细讲解，只对 Subversion 的设置进行介绍。

10.8.1 Subversion 简介

Subversion 是一款非常流行的开源版本控制系统，将来会逐渐替代 CVS，支持可在本地访问或通过网络访问的数据库和文件系统存储库。Subversion 不但提供了常用的比较、修补、标记、提交和分支等功能，其还增加了追踪移动和删除等功能。此外，Subversion 支持非 ASCII 文本和二进制数据，所有这一切都使 Subversion 不仅对传统的编程任务非常有用，同时也适于

Download at http://www.pin5i.com/

Web 开发、图书创作和其他在传统方式下未采纳版本控制功能的领域。Subversion 具有如下特性。

- 保留大多数 CVS 特性。Subversion 意味着比 CVS 更好，其拥有 CVS 的大多数特性，一般说来，Subversion 的接口与 CVS 的十分相似，除了一些强制性因素而选择其他方式。
- 目录、重命名和文件 meta-data 都已经版本化。Subversion 不仅仅对文件内容和文件存放位置进行版本控制，也对目录、副本、重命名操作进行版本控制。它也允许对附加在任意文件和目录上的 metadata 进行版本化，并提供一种机制对文件的执行权限进行控制。
- 提交成为真实的原子操作。在整体提交成功之前不会有部分提交起作用，修订号对应每次提交而不是对应每个文件，log 信息被到修订号上，并还是像 CVS 那样需要额外的位置存放。
- 可以选择 Apache 网络服务器，使用 WebDAV/DeltaV 协议。Subversion 可以使用基于 HTTP 的 WebDAV/DeltaV 协议来进行网络通信，由 Apache 网络服务器提供源代码库端的网络服务。这使得 Subversion 具有优于 CVS 的协同性，同时也提供了许多关键特性：认证、数据压缩和基本的库浏览。
- 独立服务器模式选项。Subversion 通过使用自定义的协议也提供了独立服务器选项，独立服务器模式可以作为 inetd 的一个服务运行，或者以守护进程模式运行，并且提供了基本的认证和授权，也可以使用 SSH 作为通道。
- 分支（Branching）和标签（Tagging）的开销更小（具有固定的时间）。分支和标签都使用底层的“复制”操作来实现。复制操作使用很小的、固定大小的空间。任何复制都是一个标签；如果你开始提交一个副本，那么它也就会成为一个分支。
- 天然的客户端/服务器，分层的库设计。Subversion 从开始就设计为客户端/服务器模式，从而免去了许多维护问题。代码由一系列结构化的模块组成，有定义良好的接口，设计为能够被其他程序调用。
- C/S 协议在两个方向上发送 diff。网络协议在两个方向传输 diff 时能够高效地使用带宽（CVS 发送 diff 从服务器到客户端，而不是从客户端到服务器）。
- 消耗和修改部分的大小成比例，而不是数据的大小。通常情况下，Subversion 操作所耗费的时间与此次操作引起的变化成比例，而不是对这个项目改变的绝对值，这个是 Subversion 版本库模型的特性之一。
- 可以选择数据库和纯文件的源代码库实现。源代码库可以使用嵌入的数据库后端创建，也可以使用定义格式的纯文件的后端创建。
- 对象链接的版本化。UNIX 用户可以在版本控制中放置对象链接。该链接会在 UNIX 的工作副本中重建，但不会在 win32 工作副本中重建。
- 高效处理二进制文件。Subversion 对于二进制文件的处理具有同处理文本文件一样的高效性，这是因为其在传输和存储连续的修订版本中使用二进制的文件交换算法。
- 可解析的输出。Subversion 所有的命令行客户端输出的内容都经过精心设计以达到易读和自动解析的目的。脚本化也具备较高的优先级。
- 本地化信息。Subversion 会根据当前本地设置使用 gettext()来显示传输错误、报告和帮助信息。



10.8.2 下载并安装 Subversion 客户端

NetBeans IDE 6.0 提供对 Subversion 客户端 1.3.x 或更高版本的支持。在使用 NetBeans 集成开发环境中的 Subversion 支持之前，需要在系统中安装有 Subversion 客户端软件。Subversion 客户端软件可以在 http://netbeans.open.collab.net/下载。根据计算机操作系统的不同，选择相应

Download at http://www.pin5i.com/

版本的命令行客户端，如图 10-28 所示。

下载完毕后，打开安装文件启动安装程序。默认情况下，Subversion 客户端软件可执行程序被安装在 UNIX 系统和 Mac OS X 系统的/usr/local/bin/文件夹下，在 Windows XP 系统中会安装在 C:\Program Files\CollabNet Subversion\文件夹下。在选择安装位置时，尽量使用默认位置，安装完成后单击“Finish”按钮关闭安装程序。

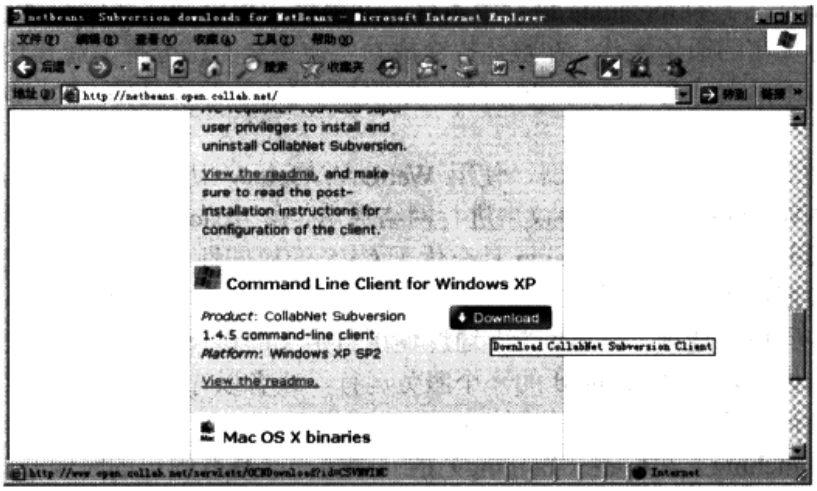


图 10-28 选择下载 Subversion 客户端软件



10.8.3 在 NetBeans IDE 中设置 Subversion

NetBeans 集成开发环境中的 Subversion 支持通过与 Subversion 客户端交互来执行版本控制命令。

NetBeans 集成开发环境会自动尝试通过使用系统的 PATH 变量来确定 Subversion 可执行文件的位置。根据不同的系统平台，以及是否将 Subversion 客户端安装到非默认位置，开发人员可能需要在 NetBeans 集成开发环境中注册 Subversion 可执行程序的路径。

在 NetBeans 集成开发环境中，选择主菜单中的“工具”→“选项”。打开“选项”对话框，选择“杂项”标签，再选择“版本控制”标签。在“版本控制系统”列表中选择“Subversion”。在“指定 SVN Home 文件夹”文本框中输入 SVN 可执行文件的路径或单击“浏览”按钮在打开的对话框中选择，如图 10-29 所示。

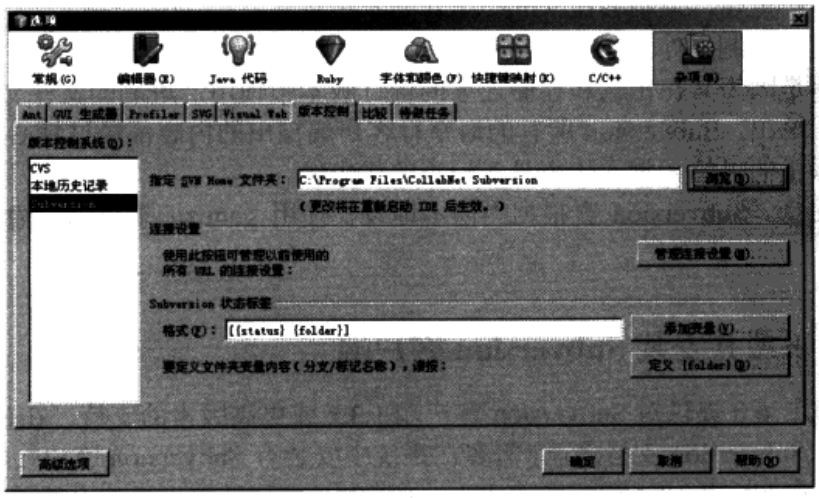


图 10-29 设置 SVN 主目录

Download at http://www.pin5i.com/

在输入路径中不要包含可执行文件，设置完毕后单击“确定”按钮。重新启动 NetBeans IDE 之后修改才能生效。

10.9 安装 Mercurial

NetBeans 除了内置 CVS 和 Subversion 版本控制系统，还支持另一款非常流行的版本控制系统——Mercurial。NetBeans 默认不支持 Mercurial，Mercurial 是作为 NetBeans 的插件提供的，可以使用下面的方法下载安装 Mercurial。

选择 NetBeans 主菜单中的“工具”→“插件”，在打开的“插件”对话框中选择“可用插件”标签，在可用插件列表中选择“Mercurial”，如图 10-30 所示。单击“安装”按钮，启动 NetBeans IDE 安装程序来安装 Mercurial。

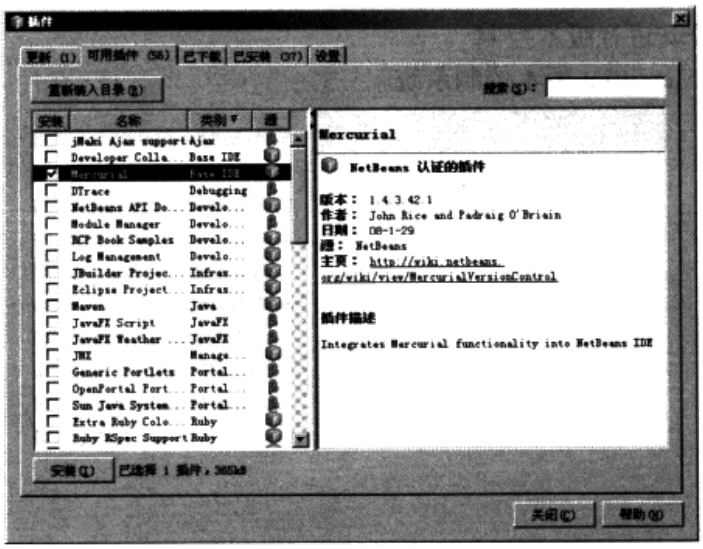


图 10-30 “插件”对话框

完成安装后会出现如图 10-31 所示的界面，安装程序询问是“立即重新启动 IDE”还是“稍后再重新启动 IDE”。如果选择立即重新启动，则 NetBeans 会重新启动以使 Mercurial 模块生效。如果选择稍后重启，则 Mercurial 暂时无法使用，在下次启动 NetBeans 时才能使用。

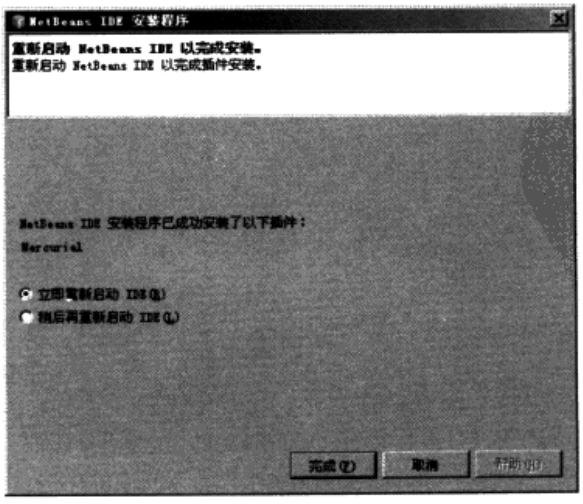


图 10-31 安装完成

Download at http://www.pin5i.com/

10.10 小结

本章主要介绍在 NetBeans IDE 中使用 CVS 进行版本控制，其中包括签入和签出、显示、比较和还原更改、更新和提交文件、查看文件历史版本记录、使用标记和分支，最后介绍了在 NetBeans 集成开发环境中如何使用其他的版本控制系统。通过本章的学习，读者应该掌握以下基本内容：

- CVS 的基本工作原理
- 签入和签出项目
- 查看文件历史版本
- 更改文件的相关操作
- 如何解决冲突
- 使用标记和分支进行版本控制
- Subversion、Mercurial 版本控制系统

Download at http://www.pin5i.com/

第 11 章

基础 Web 应用程序开发

Web 应用程序是为互联网编写的应用程序，很多语言都支持 Web 程序的开发，例如 Perl、ASP.NET、Java 等。NetBeans 集成开发环境提供了对 Web 程序开发的良好支持，可以创建空 Web 应用程序项目、快速编写 HTML 页面、建立 JSP 文件、开发 Servlet、使用 JavaServer Faces 和 Struts 框架、开发 Applet 小程序、创建和使用标记等。

本章首先介绍如何开发一个可以在网页上运行的小程序，然后介绍如何开发一个包含 HTML 页面和 JSP 页面的 Web 应用程序。

在学习本章时，假定读者已经具有了编写 JSP 程序的能力，这里着重介绍如何在 NetBeans 集成开发环境中使用这些技术进行开发。对于一些初学者，建议首先学习必要的知识。另外，为了便于初学者学习，对于讲到的一些技术，书中都进行了必要的解释，这样可以保证读者能够顺利完成本章的学习。

本章主要内容

- Applet 小程序的创建和使用
- 如何创建一个 JSP 应用程序
- 各个 HTML 组件的详细使用方法
- 各个 HTML 表单组件的详细使用方法

本章亮点

- Applet 小程序运行时问题的处理
- 在 JSP 和 JavaBean 中访问 Java DB 数据库



11.1 开发 Applet

Applet，又叫小程序，是用 Java 语言编写的一些小应用程序，这些程序直接嵌入到页面中运行。Applet 是最早可以在 Web 页面上运行的 Java 技术，能同时处理图像、声音、动画等多媒体数据。通过 Applet 可以生成有良好人机交互界面的动态 Web 页面。当支持 Java 的浏览器（例如 IE 浏览器）访问包含 Applet 的页面时，Applet 被下载到用户的计算机上执行，因此 Applet 的执行速度不受网络带宽的影响。



11.1.1 创建小程序

NetBeans 支持 Applet 的创建和开发。在 NetBeans 集成开发环境中，既可以创建一个不带可视化编辑界面的 Applet，也可以创建一个能通过窗体编辑器进行编辑的 Applet。下面分别进行介绍。

1．创建不带可视化编辑界面的 Applet

下面的步骤将创建一个简单的 Applet 程序。

Download at http://www.pin5i.com/

❶ 创建一个 Java 标准应用项目，项目的名字为“AppletDemo”。创建项目时，不创建主类。

❷ 把创建的项目设置为主项目。

❸ 选择 NetBeans 主菜单中的“文件”→“新建文件”命令，打开“新建文件”对话框。在对话框的“类别”列表中选择“Java”节点，显示如图 11-1 所示的界面。

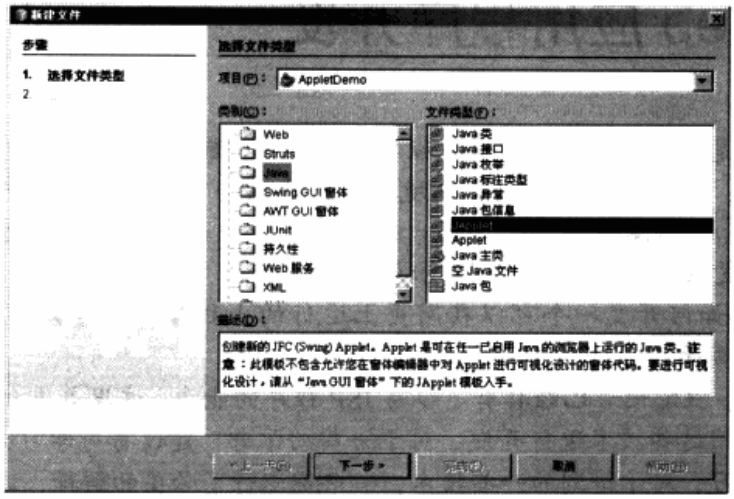


图 11-1 “新建文件”对话框

❹ 这里，可以创建两种类型的 Applet。如果在“文件类型”列表中选中“JApplet”节点，那么会创建一个新的 JFC（Swing）Applet，即 Applet 将继承于 javax.swing.JApplet 类；如果在“文件类型”列表中选中“Applet”节点，那么会创建一个新的 AWT（Abstract Window Toolkit，抽象窗口工具包）Applet，即 Applet 将继承于 java.applet.Applet 类。在“文件类型”列表中选中“JApplet”节点，单击“下一步”按钮，打开“新建 JApplet”对话框，如图 11-2 所示。

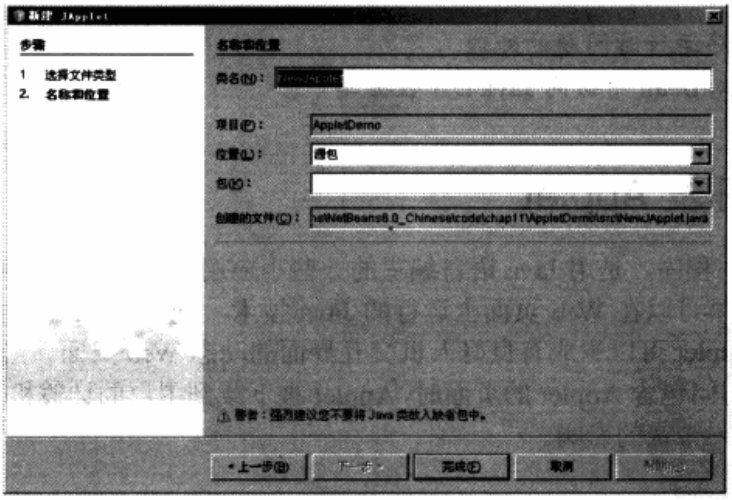


图 11-2 “新建 JApplet”对话框



说明：前面讲的 Applet 是小程序的统称。下面为了区分小程序是从哪个类派生出来的，把从 Applet 类派生出来的小程序叫做 Applet，把从 JApplet 派生出来的小程序叫做 JApplet。

❺ 在“类名”文本框中输入 JApplet 的类名“MyJApplet”，单击“完成”按钮，创建一个

Download at http://www.pin5i.com/

Applet，代码如下所示。

```
/*
 * MyJApplet.java
 *
 * Created on 2007 年 4 月 10 日, 下午 8:01
 *
 */

/**
 *
 * @author Liu Bin
 */
public class MyJApplet extends javax.swing.JApplet {

    /** Creates a new instance of MyJApplet */
    public MyJApplet() {
    }

}
```

❻ 这里创建了一个空白的 JApplet，为了能够显示内容，添加如下的方法，在屏幕上绘制文本“Hello world!”。

```
public void paint(Graphics g) {
    g.drawString("Hello world!", 50, 25);
}
```

❼ 添加引用“import java.awt.Graphics;”，完成一个简单 Applet 的创建。如果还需要添加任何功能，可以按照 Applet 的编程要求进行，这里不再赘述。

2．创建能通过窗体编辑器进行编辑的小程序

下面在上一个项目的基础上，创建一个带可视化编辑界面的 JApplet，并添加一些简单的功能。

❶ 在上一个项目的基础上，选择 NetBeans 主菜单中的“文件”→“新建文件”命令，打开“新建文件”对话框。在对话框的“类别”列表中选择“Swing GUI 窗体”节点，显示如图 11-3 所示的界面。

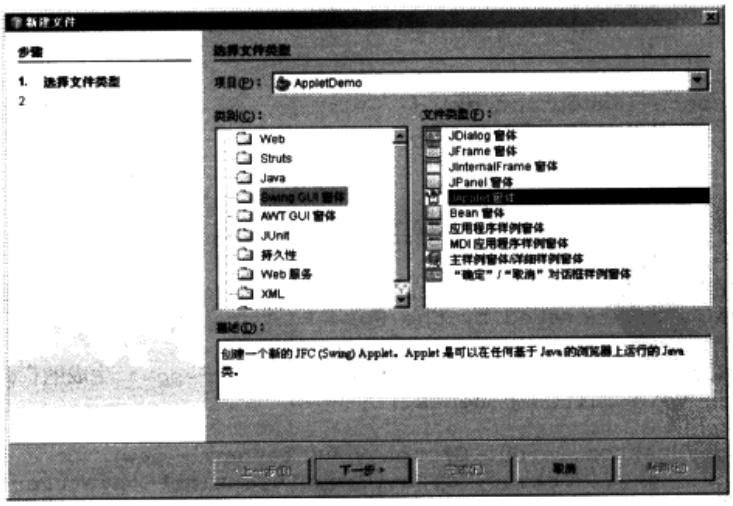


图 11-3 “新建文件”对话框

❷ 在“文件类型”列表中选择“JApplet 窗体”节点，单击“下一步”按钮，打开“新建 JApplet 窗体”对话框，如图 11-4 所示。

Download at http://www.pin5i.com/

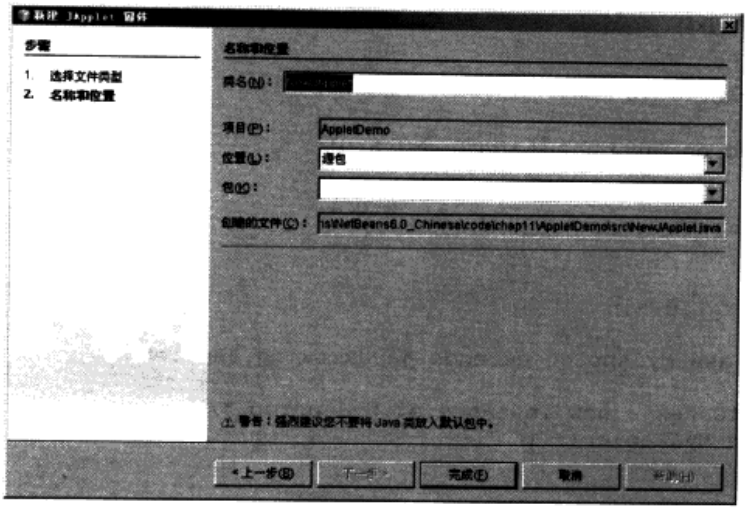


图 11-4 “新建 JApplet 窗体”对话框

❸ 在“类名”文本框中输入新 Applet 的名字“MyJAppletForm”，单击“完成”按钮，即可创建 JApplet 小程序，创建的代码如下所示。

```
/*
 * MyJAppletForm.java
 *
 * Created on 2007年4月10日, 下午8:11
 */

/**
 *
 * @author  Liu Bin
 */
public class MyJAppletForm extends javax.swing.JApplet {

    /** Initializes the applet MyJAppletForm */
    public void init() {
        try {
            java.awt.EventQueue.invokeAndWait(new Runnable() {
                public void run() {
                    initComponents();
                }
            });
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    /** This method is called from within the init() method to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc=" 生成的代码 ">
    private void initComponents() {

        org.jdesktop.layout.GroupLayout layout = new
                org.jdesktop.layout.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(
                org.jdesktop.layout.GroupLayout.LEADING)
            .add(0, 400, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
```

Download at http://www.pin5i.com/


```
            layout.createParallelGroup(
                org.jdesktop.layout.GroupLayout.LEADING)
            .add(0, 300, Short.MAX_VALUE)
        );
    }// </editor-fold>


    // 变量声明 - 不进行修改
    // 变量声明结束

}
```

❹ 在窗体中添加一个 JLabel 标签和一个 JButton 按钮，如图 11-5 所示。

❺ 单击窗体编辑器工具栏中的“连接模式”按钮，再依次单击 JButton 和 JLabel 按钮，打开“连接”向导中的“选择源事件”对话框，如图 11-6 所示。

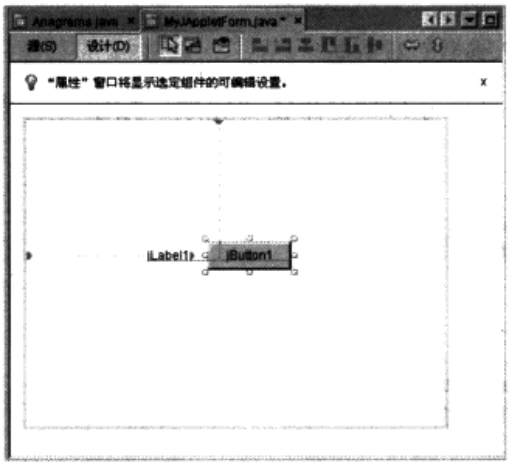


图 11-5 在 JApplet 窗体中添加组件

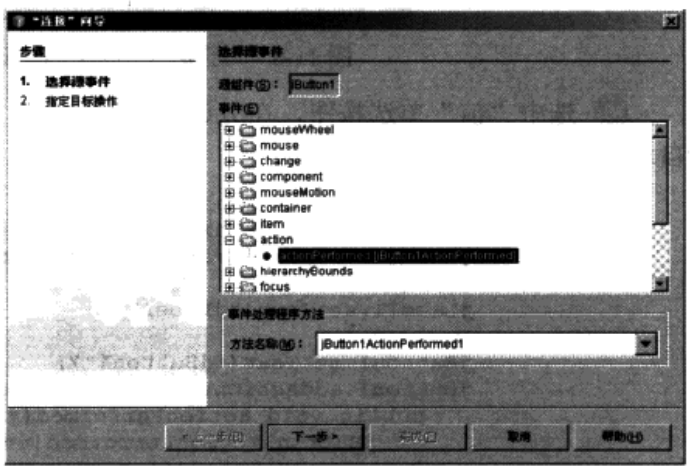


图 11-6 “连接”向导的“选择源事件”对话框

❻ 在“事件”列表中选择“action”→“actionPerformed”节点，单击“下一步”按钮，打开“连接”向导的“指定目标操作”对话框，如图 11-7 所示。

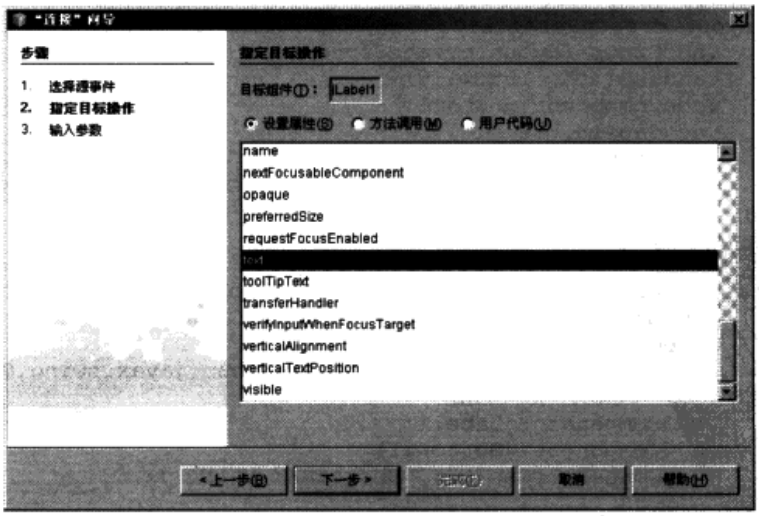


图 11-7 “连接”向导的“指定目标操作”对话框

❼ 在列表中选择“text”选项，单击“下一步”按钮，打开“连接”向导中的“输入参数”对话框，如图 11-8 所示。

Download at http://www.pin5i.com/

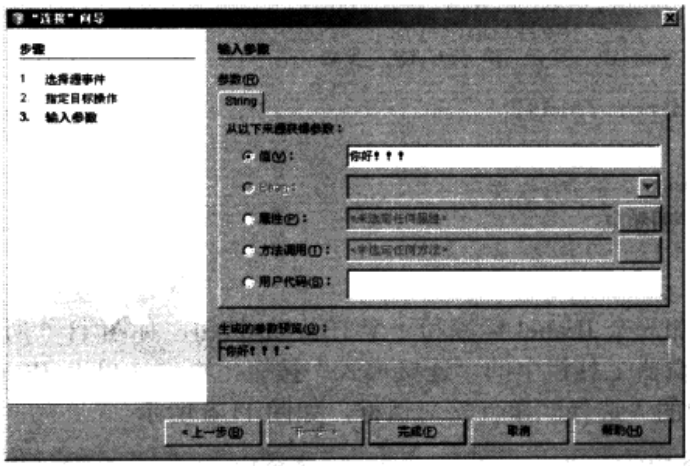


图 11-8 “连接”向导中的“输入参数”对话框

❽ 选中“值”单选按钮，在旁边的文本框中输入“你好!!!”。单击“完成”按钮，系统自动添加如下的代码。

```
    private void initComponents() {
        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();

        jLabel1.setText("jLabel1");

        jButton1.setText("jButton1");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

         javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContent
         Pane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment. LEADING)
            .addGroup(layout.createSequentialGroup()
            .addGap(115, 115, 115)
            .addComponent(jLabel1)
            .addGap(18, 18, 18)
            .addComponent(jButton1)
            .addContainerGap(147, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment. LEADING)
            .addGroup(layout.createSequentialGroup()
            .addGap(120, 120, 120)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align
                  ment.BASELINE)
            .addComponent(jLabel1)
            .addComponent(jButton1))
            .addContainerGap(153, Short.MAX_VALUE))
        );
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        jLabel1.setText("你好!!! ");
    }
```

至此，创建了一个简单的带窗体的 Applet。

Download at http://www.pin5i.com/



11.1.2 运行小程序

在 NetBeans 集成开发环境中可以方便地运行小程序，而不必编写包含 Applet 标签的 HTML 页面。下面介绍如何运行上一小节中创建的两个小程序。

在“项目”窗口或者“文件”窗口中选中要运行的小程序的 Java 文件，然后选择“运行”→ “运行文件”命令，或者按下快捷键 Shift + F6，即可打开小程序查看器运行小程序。图 11-9 是 MyJApplet 小程序的运行界面；图 11-10 是 MyJAppletForm 小程序的运行界面。



图 11-9 MyJApplet 小程序的运行界面

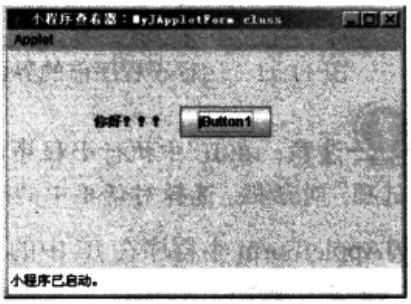


图 11-10 MyJAppletForm 小程序的运行界面

在 Windows 中，通过文件管理器打开项目目录的 build 子目录。图 11-11 是在 NetBeans 中运行小程序前的界面，图 11-12 是运行小程序后的显示界面。通过比较可以看到，NetBeans 在运行小程序时自动生成了两个 HTML 文件。打开 MyJAppletForm.html 文件，内容如下所示：

```
<HTML>
<HEAD>
   <TITLE>Applet HTML 页面</TITLE>
</HEAD>
<BODY>

<!--
*** 生成的 applet HTML 启动程序 - 不要在 "BUILD" 文件夹中进行编辑 ***

如果需要修改此 HTML 启动程序文件（例如，添加 applet 参数），
请将其复制到 SRC 文件夹中 applet 类的存放位置。如果执行了此操作，
IDE 将在您运行或调试 applet 时使用该文件。

提示：要从 JAR 文件中排除 HTML 启动程序，请在“项目属性”
对话框的“包装”页中使用排除过滤器。

有关详细信息，请参见联机帮助。
-->

<H3><HR WIDTH="100%">Applet HTML 页面<HR WIDTH="100%"></H3>

<P>
<APPLET codebase="classes" code="MyJAppletForm.class" width=350 height=200>
</APPLET>
</P>

<HR WIDTH="100%"><FONT SIZE=-1><I>由 NetBeans IDE 生成</I></FONT>
</BODY>
</HTML>
```

原来，NetBeans 会自动生成包含 Applet 标签的 HTML 页面。因此，可以直接使用这个自动生成的 HTML 页面，而不必自己手工编写。除了使用小程序查看器运行小程序外，还可以在支持 Java 的浏览器中运行小程序，即在 IE 浏览器中打开 MyJAppletForm.html 文件，就会运

Download at http://www.pin5i.com/

行 MyJApplet 小程序。

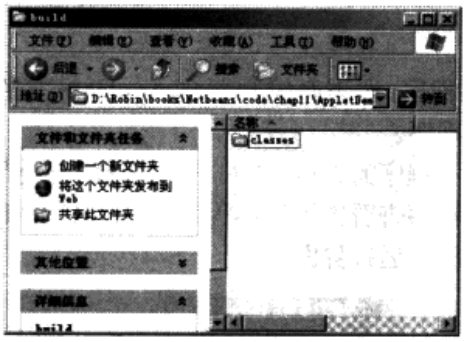


图 11-11 运行小程序前的界面

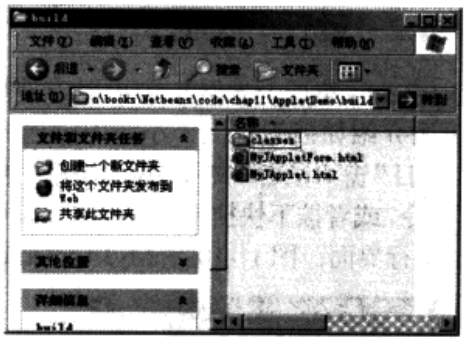


图 11-12 运行小程序后的显示界面



注意：在 IE 中执行小程序前，需要首先在 IE 中选择“工具”→“选项”命令，打开“选项”对话框。选择对话框中的“高级”标签，如图 11-13 所示。这里要确认选中使用 JRE。

MyJAppletForm 小程序在 IE 中的运行界面如图 11-14 所示。



图 11-13 确认选中使用 JRE

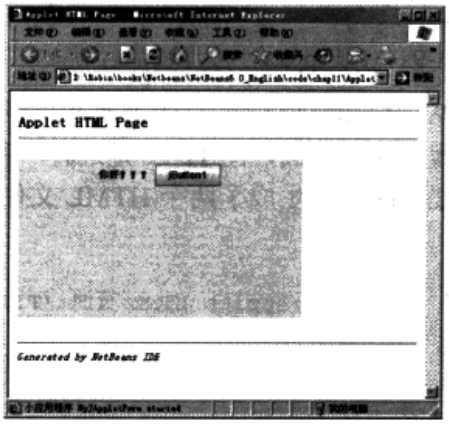


图 11-14 MyJApplet 小程序运行界面



11.1.3 在 JDK 1.5 中运行小程序

当 JDK 的版本为 1.5 时，如果在 IE 中运行 MyJAppletForm 小程序，就会出现如图 11-15 所示的界面。

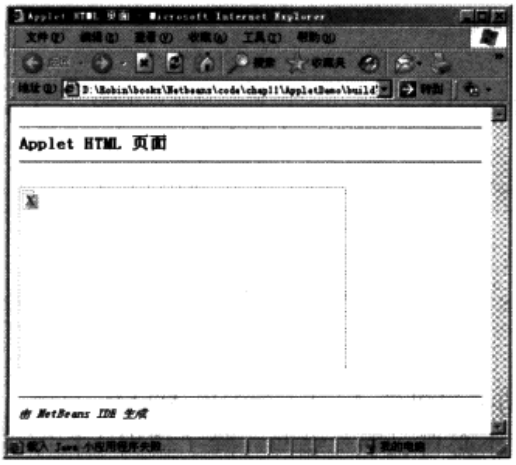


图 11-15 MyJAppletForm 小程序运行界面

Download at http://www.pin5i.com/

在小程序上单击鼠标右键，选择弹出菜单中的“打开 Java 控制台”命令，打开 Java 控制台，如图 11-16 所示。从控制台显示的信息中可以看到，主要原因是没有找到 org.jdesktop.layout.GroupLayout 类。原来，org.jdesktop.layout.GroupLayout 类在 JDK 1.5 中不是标准的布局类，它包含在 NetBeans 安装目录的 platform6\modules\ext\swing-layout-1.0.jar 文件中。

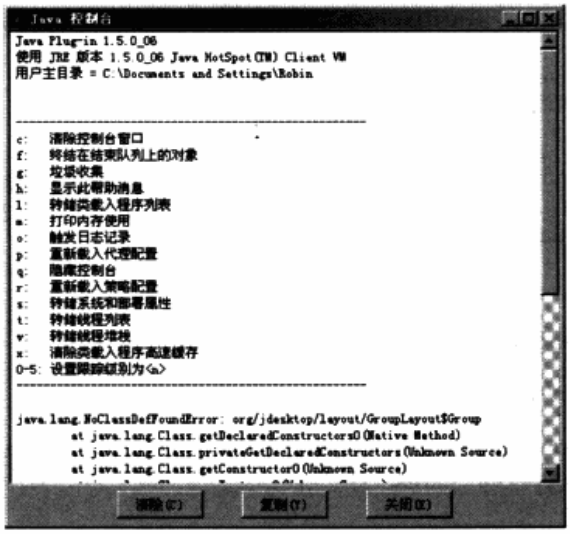


图 11-16 Java 控制台

说明：在 JDK 1.6 中，org.jdesktop.layout.GroupLayout 类已经变为标准的布局了，因此不会出现这种问题。这里建议读者在安装 NetBeans 6.0 时，要首先安装 JDK 1.6。

要解决这个问题，可以把 org.jdesktop.layout.GroupLayout 类添加到类路径中，也可以将 GroupLayout 布局替换为 FlowLayout 布局。在光盘的 code\chap11\AppletDemo\src 目录中，打开 MyJAppletForm2.java 文件，其中代码已经修改为使用 FlowLayout 布局：

```
/*
 * MyJAppletForm.java
 *
 * Created on 2007年4月10日, 下午8:11
 */

/**
 *
 * @author  Liu Bin
 */
public class MyJAppletForm extends javax.swing.JApplet {

    /** Initializes the applet MyJAppletForm */
    public void init() {
        try {
            java.awt.EventQueue.invokeAndWait(new Runnable() {
                public void run() {
                    initComponents();
                }
            });
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    /** This method is called from within the init() method to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
```

Download at http://www.pin5i.com/


```
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc=" 生成的代码 ">
    private void initComponents() {
        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();

        getContentPane().setLayout(new java.awt.FlowLayout());

        jLabel1.setText("jLabel1");
        getContentPane().add(jLabel1);

        jButton1.setText("jButton1");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        getContentPane().add(jButton1);

    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        jLabel1.setText("你好!!! ");
    }


    // 变量声明 - 不进行修改
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    // 变量声明结束

}
```

修改以后，再次运行程序，显示小程序的运行界面如图 11-17 所示。

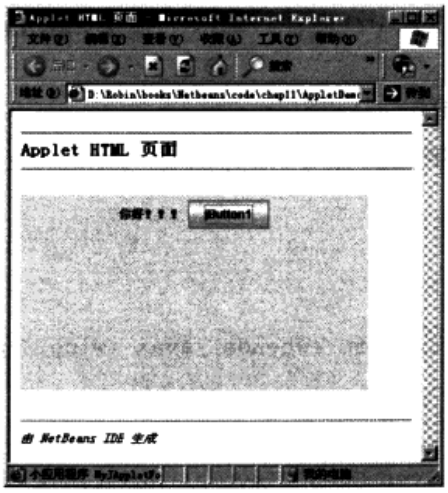


图 11-17　修改布局以后的 MyJAppletForm 运行界面

11.2　创建一个简单的 Web 应用程序

NetBeans 中可以快速创建基于 Java 技术的 Web 应用程序。NetBeans 集成开发环境支持 JSP 2.0、Servlet 2.4、JSTL 1.1、JavaServer Pages 规范 2.0 版等技术。下面首先介绍如何创建一个简单的 Web 应用程序项目。

Download at http://www.pin5i.com/

11.2.1 创建 Web 应用程序项目

选择 NetBeans 主菜单中的“文件”→“新建项目”命令，打开“新建项目”对话框。在该对话框的“类别”列表中选择“Web”节点，如图 11-18 所示。

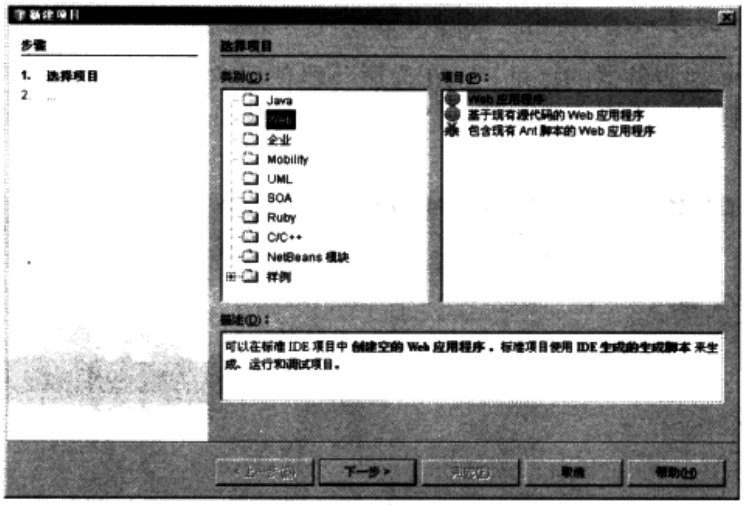


图 11-18 “新建项目”对话框

在“项目”列表中选择“Web 应用程序”节点，单击“下一步”按钮，打开“新建 Web 应用程序”对话框，用于新建包含空 Web 应用程序的标准项目，如图 11-19 所示。

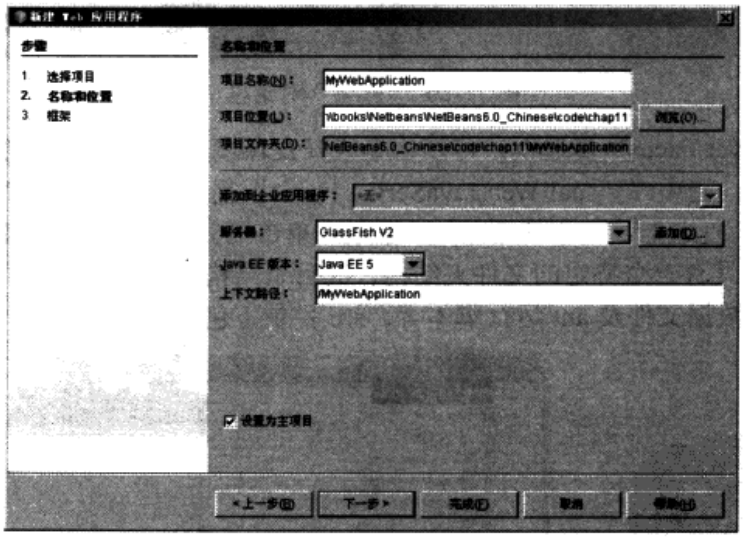


图 11-19 “新建 Web 应用程序”对话框

在“项目名称”文本框中输入项目的显示名称，此名称还用于创建项目文件夹和上下文路径。“项目位置”文本框用于指定要存储项目元数据和源代码的位置。单击“浏览”按钮，可以选择不同的目录。

“项目文件夹”文本框中显示了要存储项目元数据和源代码的文件夹。此文件夹是由项目位置和项目名称组合而成的，不能进行修改。

“添加到企业应用程序”下拉列表框可以选择现有的企业应用程序，这样新建的项目将作为模块添加到企业应用程序中。

Download at http://www.pin5i.com/

“服务器”下拉列表框用于指定 Web 应用程序的服务器，此处只能选择在集成开发环境中注册的服务器。

“Java EE 版本”下拉列表框用于指定运行 Web 应用程序时所使用的 J2EE 级别。“上下文路径”文本框用于指定访问该 Web 应用程序的路径，即 Web 应用程序内容的 URL 名称空间。例如属性值设置为/MyWebApplication，则项目中 index.jsp 文件的访问 URL 为：

```
http://hostname:port/MyWebApplication/index.jsp
```

其中，hostname 为服务器的地址，port 为服务器使用的端口号。选中“设置为主项目”复选框，将新建的项目设置为主项目。单击“下一步”按钮，打开“框架”对话框，如图 11-20 所示。

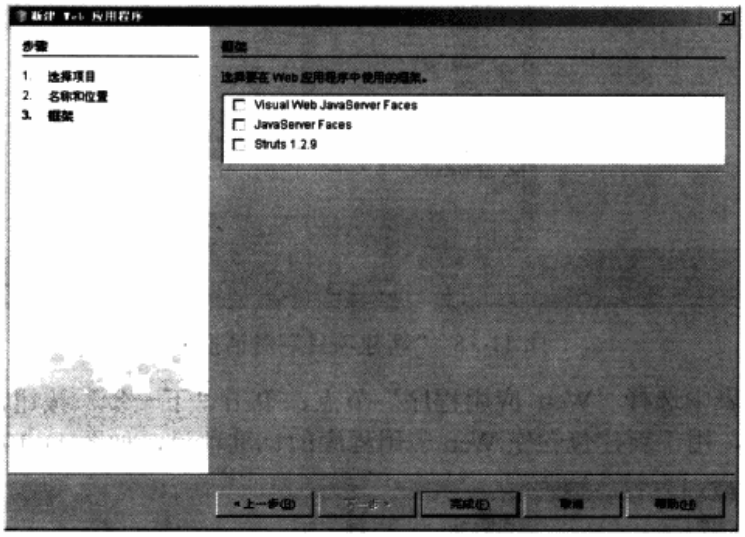


图 11-20 “框架”对话框

选中“JavaServer Faces”复选框，可以让集成开发环境为应用程序提供使用 JSF 框架所需的所有源文件和库；选中“Visual Web JavaServer Faces”复选框，将支持可视化的 JSF 组件；选中“Struts 1.2.9”复选框，则会使用 Struts 框架。单击“完成”按钮，创建 Web 应用程序。

Web 应用程序具有一个特定的文件夹结构，其目录结构如图 11-21 所示。其中，nbproject 目录存放项目相关数据文件及 ant 运行脚本等。src 目录下包含以下两个文件夹。

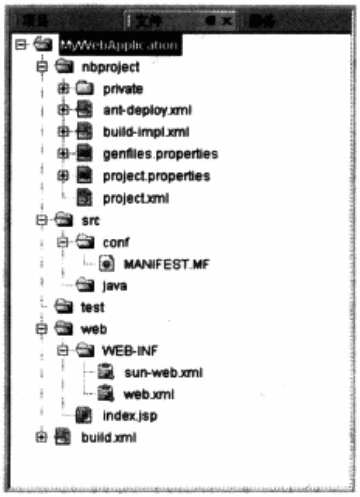


图 11-21 Web 应用程序的目录结构

Download at http://www.pin5i.com/

- java 文件夹：存放 Java 源代码，例如 Servlet 文件。
- conf 文件夹：配置文件夹，里面包含 MANIFEST.MF 文件。

test 目录中可以包含单元测试程序；web 目录包含 Web 页面文件，例如 JSP 页面文件、HTML 页面文件等。在其下包含一个 WEB-INF 子目录。

WEB-INF 目录中包含 web.xml 文件，该文件是部署描述符文件，用于配置属于 Web 应用程序的组件，如 Servlet 和 JSP 文件。这个文件必须存在。在创建 Web 应用程序时，如果选择的服务器是 Sun Java System Application Server，那么还会创建一个 sun-web.xml 文件。该文件是特定于服务器的部署描述符，用于配置 Sun Java System Application Server。



11.2.2 运行 Web 应用程序

在创建 Web 应用程序时，系统会自动地创建一个 index.jsp 文件，其内容如下所示：

```
<%--
    Document   : index
    Created on : 2007-12-30, 23:08:38
    Author     : Liu Bin
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h2>Hello World!</h2>
    </body>
</html>
```

其中，<%@page contentType="text/html" pageEncoding="UTF-8"%>用于说明 JSP 文件的 MIME 类型和编码，这里指定 MIME 类型为“text/html”类型；页面的编码格式为 UTF-8 编码。如果要支持简体中文，可以使用 GB2312、GB18030 或者 GBK（汉字内码扩展规范）；如果要支持繁体中文，可以使用 Big5。



说明：关于编码的具体说明，请读者参考 Sun 网站的具体说明 http://java.sun.com/j2se/ 1.5.0/ docs/guide/ intl/encoding.doc.html。

在 JSP 页面编写完成后，在“项目”窗口中右键单击 JSP 文件，然后从弹出的菜单中选择“编译文件”命令（快捷键为 F9），可以编译单个文件。单击工具栏上的“运行主项目”按钮，可以运行 Web 应用程序，运行界面如图 11-22 所示。

在项目创建以后，也可以修改运行的目标服务器。在“项目”窗口中，右键单击 Web 项目节点，选择弹出菜单中的“属性”命令，打开“项目属性”对话框中。选择“运行”节点，设置应用程序的运行属性，如图 11-23 所示。在“服务器”下拉列表框中，选择要运行的服务器名称。“上下文路径”文本框可以用来修改应用程序的上下文路径。修改上下文路径后，需要使用新的上下文路径才能启动应用程序。如果还需要指定相对路径，则可以在“相对 URL”文本框中进行输入。

Download at http://www.pin5i.com/



图 11-22 MyWebApplication 的运行界面

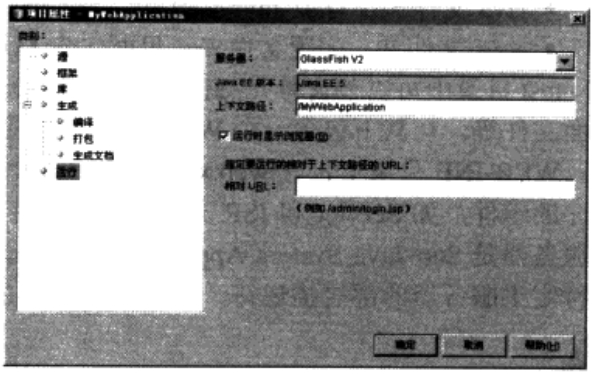


图 11-23 设置 Web 应用程序的运行参数

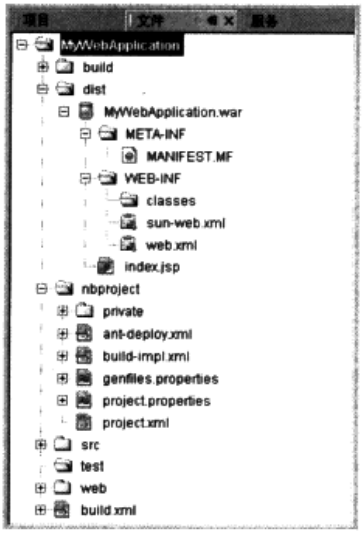


图 11-24 MyWebApplication 项目的 WAR 文件的结构

11.2.3 调试 Web 应用程序

可以在 NetBeans 集成开发环境中使用 Sun Microsystems JPDA 调试器来调试 Web 应用程序、Servlet 和 JSP 文件。如同调试 Java 程序一样，可以在集成开发环境中设置新的监视变量，将光标放在变量上查看变量的值，并在 JSP 文件、JSP 文档和 JSP 片段中设置断点。在调试引用标记文件的 JSP 文件时，还将会调试该标记文件。关于程序的调试，已经在第 3 章中进行了详细介绍，这里不再赘述。

11.2.4 生成 Web 应用程序

在“项目”窗口中右键单击项目节点，选择弹出菜单中的“生成项目”命令或者“清理并生成项目”命令，NetBeans 都会把应用程序包装为 Web 应用程序归档（Web Application Archive, WAR）文件。生成的 WAR 文件保存在项目文件夹的 dist 子文件夹中。对于 MyWebApplication 项目，WAR 文件的结构如图 11-24 所示。

如果要删除 WAR 文件，则可以在“文件”窗口中右键单击要删除的 WAR 文件，然后选择弹出菜单中的“删除”命令。

如果要修改 WAR 文件的默认输出路径，则在项目文件夹中，打开 nbproject 子文件夹中的 project.properties 文件，修改如下三行内容：

```
dist.dir=dist
war.name=MyWebApplication.war
dist.war=${dist.dir}/${war.name}
```

其中，dist.dir 属性用于定义 WAR 文件的完整路径；war.name 属性指定了 WAR 文件的名称；dist.war 属性由 dist.dir 属性和 war.name 属性构成。

如果要在 WAR 中添加其他的资源，在“项目”窗口中，右键单击项目节点，然后选择弹出菜单中的“属性”命令。在打开的“项目属性”对话框中，选择“生成”→“打包”节点，如图 11-25 所示。

在“不包含在 WAR 文件中”文本框中，输入不希望包含在 WAR 文件中的文件类型通配符。如果要压缩 WAR 文件，则要选中“压缩 WAR 文件”复选框。单击下列任意按钮，可以添加期望的内容。

Download at http://www.pin5i.com/

- “添加文件/文件夹”按钮：添加单独的文件，或者某个文件夹。
- “添加库”按钮，单击该按钮，打开“添加库”对话框，选择添加系统中的库。
- “添加项目”按钮：添加另外的项目。

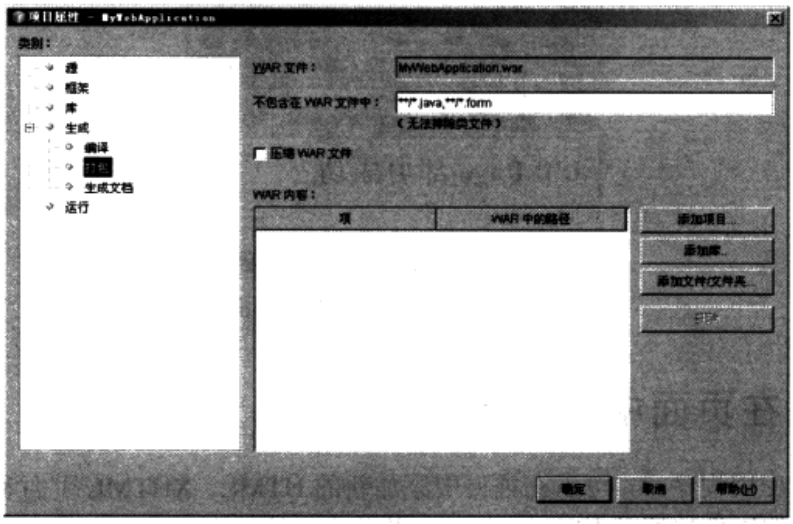


图 11-25 添加打包资源

11.2.5 部署 Web 应用程序

所谓部署 Web 应用程序，就是把已经开发的 Web 应用程序放置到服务器中。NetBeans 提供了快速部署 Web 应用程序的方法。下面通过一个例子来演示 Web 应用程序的部署过程。

❶ 在前面创建的 MyWebApplication 项目中，打开 index.jsp 文件，把下面的第 1 行代码修改为第 2 行代码：

```
<h2>Hello World!</h2>
<h2>JSP Page 部署成功</h2>
```

❷ 在“项目”窗口中，右键单击项目节点，选择弹出菜单中的“部署项目”命令，开始进行部署。对于一些服务器，如果设置了管理口令，那么第一次运行部署任务时会显示登录对话框。在输入正确的管理员账号和密码后才能继续部署。关于服务器的管理和配置，请读者参考第 6 章的介绍。

❸ 部署成功后，在“输出”窗口中会显示如下的提示信息。

```
init:
deps-module-jar:
deps-ear-jar:
deps-jar:
library-inclusion-in-archive:
library-inclusion-in-manifest:
compile:
compile-jsps:
do-dist:
dist:
正在取消部署...
OK - Undeployed application at context path /MyWebApplication
就地在 C:\Documents and Settings\Robin\桌面\MyWebApplication\build\web 处部署
OK - Deployed application at context path /MyWebApplication
OK - Started application at context path /MyWebApplication
run-deploy:
生成成功（总时间：17 秒）
```

Download at http://www.pin5i.com/

❹ 打开 IE 浏览器，输入地址 http://localhost:8084/MyWebApplication/，回车后显示如图 11-26 所示的界面。可见，此时的 JSP 页面已经和原来如图 11-22 所示的页面不同了，这说明修改后的 JSP 文件已经被成功部署到服务器了。



图 11-26 修改后的 JSP 页面

11.3 在页面中使用 HTML 组件

在创建 Web 应用程序后，可以在项目中添加新的 HTML、XHTML 页面和 JSP 页面，并利用 NetBeans 提供的组件快速开发 Web 页面。本节首先介绍标准 HTML 组件的使用方法以及 JSP 文件的编写方法，然后在此基础上介绍一个登录模块的全部开发过程。

11.3.1 创建 HTML 和 XHTML 页面

在创建 Web 应用程序后，选择 NetBeans 主菜单中的“文件”→“新建文件”命令。在打开的“新建文件”对话框中，选择“类别”列表中的 Web 节点，如图 11-27 所示。

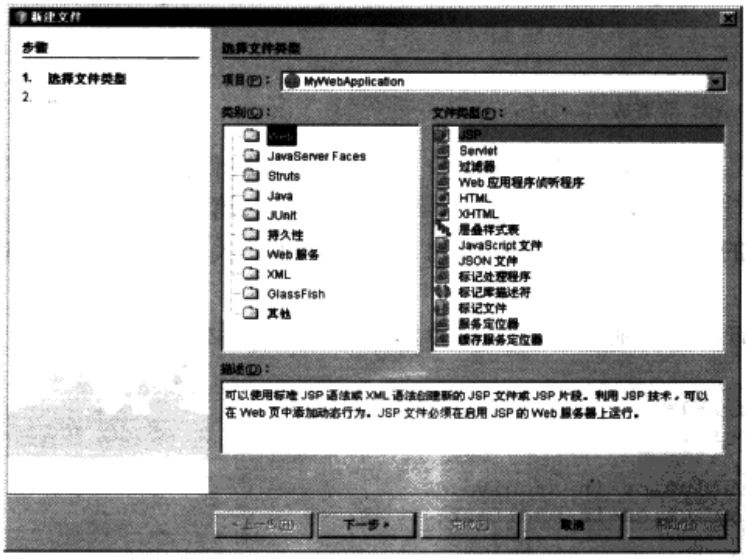


图 11-27 新建 Web 文件

在这个对话框中，可以选择创建 HTML 文件或者 XHTML 文件。这两个文件的创建过程比较类似，下面以创建 HTML 文件为例进行讲解。

在“文件类型”列表中选择“HTML”节点，单击“下一步”按钮，打开“新建 HTML 文件”对话框，如图 11-28 所示。在“HTML 文件名”文本框中输入 HTML 文件的名字。默认情况下 HTML 文件会包含在 Web 目录中，如果需要把该文件放在源代码目录中，可以在“位置”下拉列表框中选择“源包”选项。

Download at http://www.pin5i.com/

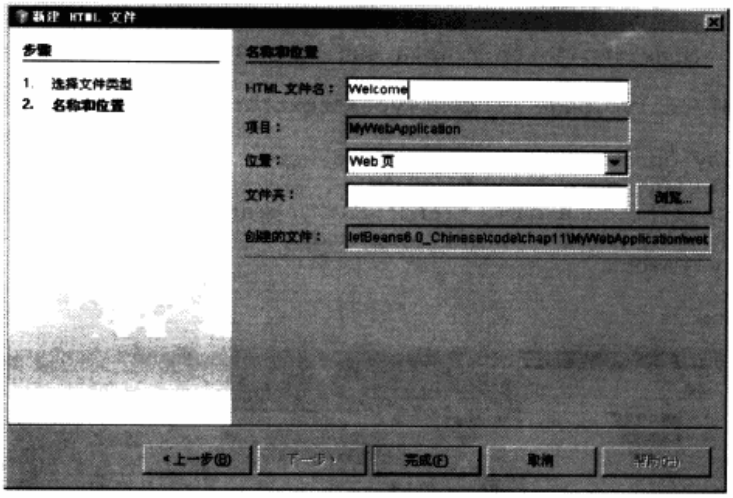


图 11-28 “新建 HTML 文件”对话框

单击“完成”按钮，创建 HTML 页面。在“项目”窗口中双击创建的 HTML 文件，在源代码编辑器中打开该文件进行编辑，代码如下所示：

```
<!--
    Document   : HTML1
    Created on : 2007-12-30, 23:29:08
    Author     : Liu Bin
-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>

  </body>
</html>
```

此时，可以从“组件面板”窗口中拖曳 HTML 组件到源代码编辑器中，对 HTML 页面进行编辑。关于 HTML 组件的使用方法请参考第 11.3.3 节和第 11.3.4 节。



11.3.2 创建 JSP 页面

在如图 11-27 所示的“新建文件”对话框中，选择“文件类型”列表中的“JSP”节点，可以创建一个新的 JSP 文件。单击“下一步”按钮，打开“新建 JSP 文件”对话框，如图 11-29 所示。

在“JSP 文件名”文本框中输入 JSP 文件的名字。默认情况下 HTML 文件会包含在 Web 目录中，因此“位置”下拉列表框中要选择“Web 页”选项。

选中“JSP 文件（标准语法）”单选按钮，则创建使用 JSP 标准语法的 JSP 文件。创建的 JSP 文件如第 11.2.2 节中的 index.jsp 文件所示。

选中“JSP 文档（XML 语法）”单选按钮，则创建使用 XML 语法的 JSP 文件。创建的 JSP 文件内容如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Document   : index2
    Created on : 2007-12-30, 23:31:23
```

Download at http://www.pin5i.com/


```
    Author     : Liu Bin
-->
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.0">

    <jsp:directive.page contentType="text/html" pageEncoding="UTF-8"/>

    <!-- any content can be specified here, e.g.: -->
    <jsp:element name="text">
        <jsp:attribute name="lang">EN</jsp:attribute>
        <jsp:body>Hello World!</jsp:body>
    </jsp:element>

</jsp:root>
```

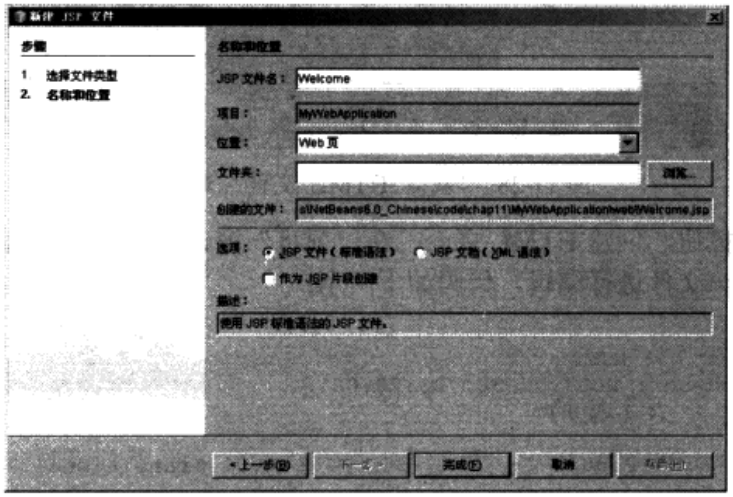


图 11-29 “新建 JSP 文件”对话框

如果选中“作为 JSP 片段创建”复选框，则使用 XML 语法，以静态方式包括来自其他页面的 JSP 文件。单击“完成”按钮，即可创建 JSP 文件。

说明： 通常把创建的 JSP 代码片段保存在 WEB-INF/jspf/文件夹下，这样就无法提供 URL 直接访问该文件。

11.3.3 HTML 组件面板

在 HTML、XHTML 或者 JSP 页面中都可以使用 HTML 语法。NetBeans 为 HTML 语法提供了标准组件，可以直接拖曳到页面中。对于初学者，可以不必详细了解 HTML 语法，也能够快速开发 Web 页面。

在图 11-30 中，左边的图是在源代码编辑器中打开 HTML 文件或者 XHTML 文件时的“组件面板”窗口；图 11-30 中右边的图是在源代码编辑器中打开 JSP 文件时的“组件面板”窗口。

HTML 标准组件包括 HTML 组件和 HTML 表单组件两个部分。在 HTML 或者 XHTML 页面中只可以添加这两个类型的组件。在 JSP 页面中还可以添加其他的一些非 HTML 组件，例如 JSTL 组件。下面主要讲解如何在 JSP 文件中添加 HTML 组件和 HTML 表单组件（在 HTML 和 XHTML 文件中的使用方法一样）。其他组件将在后续几节中进行介绍。

11.3.4 使用 HTML 表

HTML 表是由 HTML 的<table></table>标签、<tr></tr>标签、<td></td>标签等几个标签组

Download at http://www.pin5i.com/

成的。在“组件面板”窗口中选中“表”组件，然后拖曳到源代码编辑器中希望出现表格的地方，如图 11-31 所示。

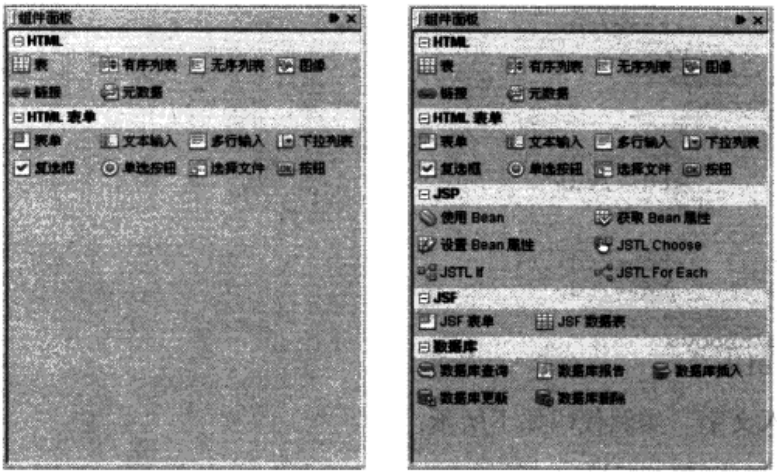


图 11-30 “组件面板”窗口

在释放鼠标后，自动打开“插入 表”对话框，如图 11-32 所示。对话框中各个属性的说明如表 11-1 所示。

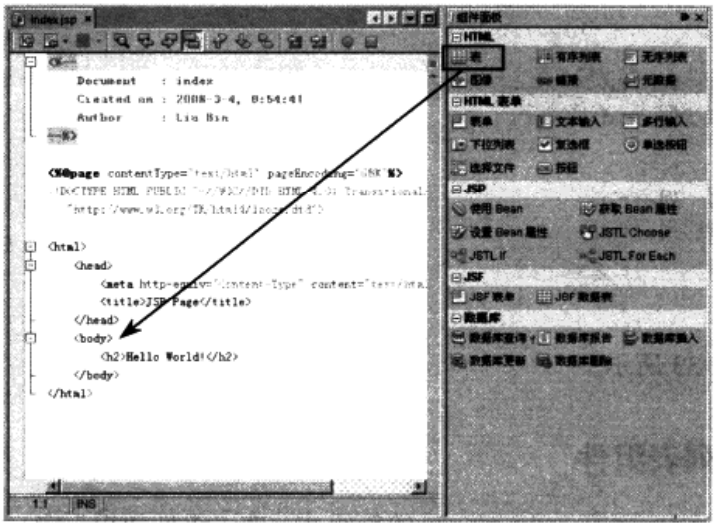


图 11-31 使用 HTML 表

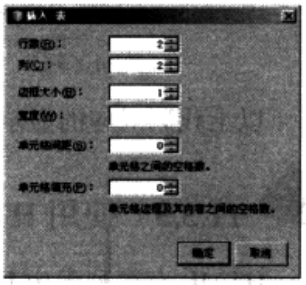


图 11-32 “插入 表”对话框

表 11-1 HTML 表属性

属 性	说 明
行数	表格的行数，对应<tr></tr>标签
列	表格的列数，对应<td></td>标签
边框大小	表格的边框大小。如果不需要边框则可以把这个属性设置为 0
宽度	表格的宽度
单元格间距	单元格之间的空格数
单元格填充	单元格边框和内容之间的空格数

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 表格。图 11-32 中的设置对应的 HTML 代码如下所示：

Download at http://www.pin5i.com/


```
<table border="1">
    <thead>
        <tr>
            <th></th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td></td>
            <td></td>
        </tr>
        <tr>
            <td></td>
            <td></td>
        </tr>
    </tbody>
</table>
```

在表格中输入文字，实例代码如下所示：

```
<table border="1">
    <thead>
        <tr>
            <th>第一列的表头</th>
            <th>第二列的表头</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>第一行，第一列</td>
            <td>第一行，第二列</td>
        </tr>
        <tr>
            <td>第二行，第一列</td>
            <td>第二行，第二列</td>
        </tr>
    </tbody>
</table>
```

以上代码对应的显示界面如图 11-33 所示。

11.3.5 使用 HTML 有序列表组件

HTML 有序列表由 HTML 的标签、标签等组成。在“组件面板”窗口中选中“有序列表”组件，然后拖曳到源代码编辑器中希望出现的地方。在释放鼠标后，自动打开“插入 有序列表”对话框，如图 11-34 所示。对话框中各个属性的说明如表 11-2 所示。



图 11-33 HTML 表格显示界面

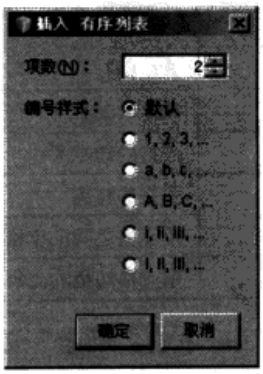


图 11-34 “插入 有序列表”对话框

Download at http://www.pin5i.com/

表 11-2 HTML 有序列表属性

属 性	说 明
项数	列表的项目数
默认	项目符号使用默认的式样
1, 2, 3, …	使用数字作为项目编号
a, b, c, …	使用小写的英文字符 a，b，c 等作为项目编号
A, B, C, …	使用大写的英文字符 A，B，C 等作为项目编号
i, ii, iii, …	使用数字作为项目编号
I, II, III, …	使用数字作为项目编号

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 有序列表。图 11-34 中的设置对应的 HTML 代码如下所示：

```
<ol>
    <li></li>
    <li></li>
</ol>
```

其他几种式样对应的 HTML 代码分别如下所示（按照图 11-34 中的顺序）：

```
<ol type="1">
    <li></li>
    <li></li>
</ol>
<p></p>
<ol type="a">
    <li></li>
    <li></li>
</ol>
<p></p>
<ol type="A">
    <li></li>
    <li></li>
</ol>
<p></p>
<ol type="i">
    <li></li>
    <li></li>
</ol>
<p></p>
<ol type="I">
    <li></li>
    <li></li>
</ol>
```

在有序列表中输入文字，代码如下所示：

```
<ol>
    <li>默认式样 - 有序列表项目 1</li>
    <li>默认式样 - 有序列表项目 2</li>
</ol>
<p></p>
<ol type="1">
    <li>数字式样 - 有序列表项目 1</li>
    <li>数字式样 - 有序列表项目 2</li>
</ol>
<p></p>
<ol type="a">
    <li>小写英文式样 - 有序列表项目 1</li>
    <li>小写英文式样 - 有序列表项目 2</li>
</ol>
```

Download at http://www.pin5i.com/


```
<p></p>
<ol type="A">
    <li>大写英文式样 - 有序列表项目 1</li>
    <li>大写英文式样 - 有序列表项目 2</li>
</ol>
<p></p>
<ol type="i">
    <li>小写罗马数字式样 - 有序列表项目 1</li>
    <li>小写罗马数字式样 - 有序列表项目 2</li>
    <li>小写罗马数字式样 - 有序列表项目 3</li>
    <li>小写罗马数字式样 - 有序列表项目 4</li>
    <li>小写罗马数字式样 - 有序列表项目 5</li>
</ol>
<p></p>
<ol type="I">
    <li>大写罗马数字式样 - 有序列表项目 1</li>
    <li>大写罗马数字式样 - 有序列表项目 2</li>
    <li>大写罗马数字式样 - 有序列表项目 3</li>
    <li>大写罗马数字式样 - 有序列表项目 4</li>
    <li>大写罗马数字式样 - 有序列表项目 5</li>
</ol>
```

以上代码对应的显示界面如图 11-35 所示。

11.3.6 使用 HTML 无序列表组件

HTML 无序列表由 HTML 的标签、标签等组成。在“组件面板”窗口中选中“无序列表”组件，然后拖曳到源代码编辑器中希望出现的地方。在释放鼠标后，自动打开“插入 无序列表”对话框，如图 11-36 所示。对话框中各个属性的说明如表 11-3 所示。

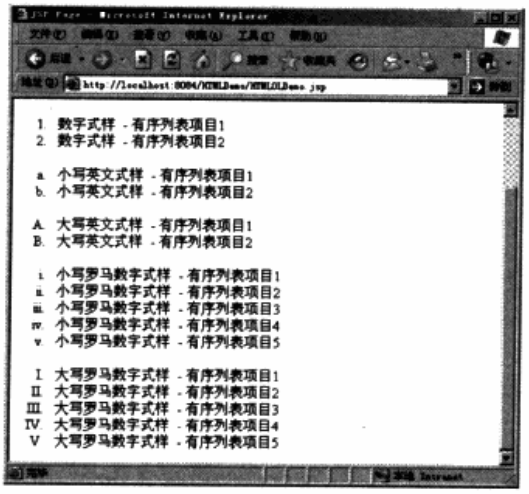


图 11-35 有序列表显示界面

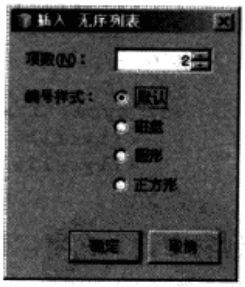


图 11-36 “插入 无序列表”对话框

表 11-3 HTML 无序列表属性

属 性	说 明
项数	列表的项目数
默认	项目符号使用默认的式样
磁盘	使用磁盘形状符号作为项目符号
圆形	使用圆形作为项目符号
正方形	使用正方形作为项目符号

Download at http://www.pin5i.com/

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 无序列表。图 11-36 中的设置对应的 HTML 代码如下所示：

```
<ul>
    <li></li>
    <li></li>
</ul>
```

其他几种式样对应的 HTML 代码分别如下所示（按照图 11-36 中的顺序）：

```
<ul type="disc">
    <li></li>
    <li></li>
</ul>
<p></p>
<ul type="circle">
    <li></li>
    <li></li>
</ul>
<p></p>
<ul type="square">
    <li></li>
    <li></li>
</ul>
```

在无序列表中输入文字，代码如下所示：

```
<ul>
    <li>默认式样 - 无序列表项目 1</li>
    <li>默认式样 - 无序列表项目 2</li>
</ul>
<p></p>
<ul type="disc">
    <li>磁盘式样 - 无序列表项目 1</li>
    <li>磁盘式样 - 无序列表项目 2</li>
</ul>
<p></p>
<ul type="circle">
    <li>圆形式样 - 无序列表项目 1</li>
    <li>圆形式样 - 无序列表项目 2</li>
</ul>
<p></p>
<ul type="square">
    <li>正方形式样 - 无序列表项目 1</li>
    <li>正方形式样 - 无序列表项目 2</li>
</ul>
```

以上代码对应的显示界面如图 11-37 所示。

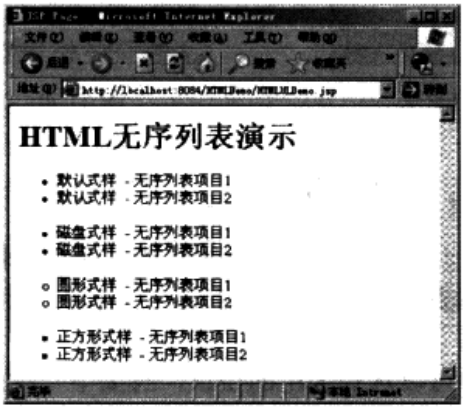


图 11-37 HTML 无序列表显示界面

Download at http://www.pin5i.com/

11.3.7 使用 HTML 图像组件

HTML 图像由 HTML 的<image></image>标签组成。在“组件面板”窗口中选中“图像”组件，然后拖曳到源代码编辑器中希望出现的地方。在释放鼠标后，自动打开“插入 图像”对话框，如图 11-38 所示。对话框中各个属性的说明如表 11-4 所示。

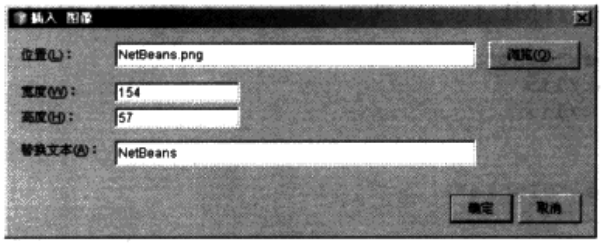


图 11-38 “插入 图像”对话框

表 11-4 HTML 图像属性

属 性	说 明
位置	用于输入要显示的图像的路径，单击“浏览”按钮，可以打开如图 11-39 所示的对话框，在项目目录的 Web 子目录中选择图像
宽度	图像的宽度
高度	图像的高度
替换文本	如果不能显示图像时，将显示替换文本的内容

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 图像。图 11-38 中的设置对应的 HTML 代码如下所示：

```
<img src="NetBeans.png" width="154" height="57" alt="NetBeans"/>
```

以上代码对应的显示界面如图 11-40 所示。

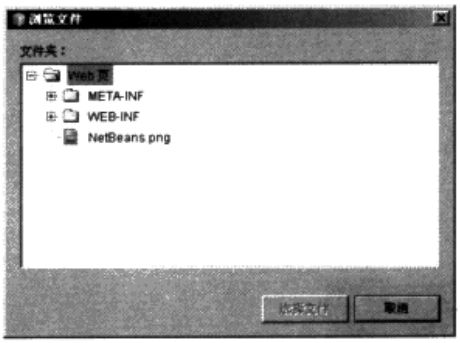


图 11-39 选择要显示的图像



图 11-40 HTML 图像显示界面

11.3.8 使用 HTML 链接组件

HTML 链接由 HTML 的 标签组成。在“组件面板”窗口中选中“链接”组件，然后拖曳到源代码编辑器中希望出现的地方。在释放鼠标后，自动打开“插入 链接”对话框，如图 11-41 所示。

在“协议”下拉列表框中可以选择使用的协议，有如下几种协议。

- file：使用文件协议，链接文件位于当前机器的文件系统中。

Download at http://www.pin5i.com/

- http：使用 HTTP 协议，链接文件通常位于另外的 Web 服务器中。
- https：使用加密的 HTTP 协议，链接文件通常位于另外的 Web 服务器中。
- ftp：使用 FTP 协议，链接文件通常位于另外的 FTP 服务器中。
- mailto：使用邮件协议，将向指定的地址发送邮件。

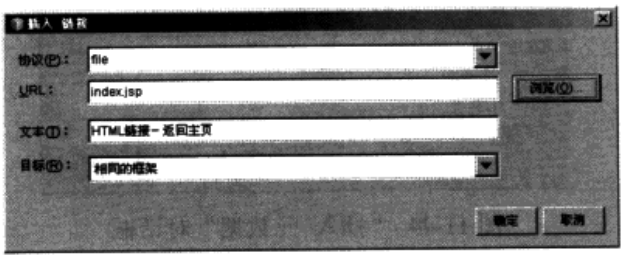


图 11-41 “插入 链接”对话框

在 URL 文本框中，输入被链接的文件，单击该链接后会打开这里设置的文件。单击“浏览”按钮，打开“浏览文件”对话框，在项目目录的 Web 子目录中选择被链接的文件，如图 11-42 所示。

在“文本”文本框中输入显示的文字，单击该文字会打开被链接的内容。“目标”下拉列表框用于选择在哪里打开新的页面，有如下几个选项。

- 相同的框架：在与当前页面相同的框架中打开新的页面。
- 新窗口：打开一个新的窗口，在新窗口中打开新的页面，当前页面并不消失。
- 父框架：在当前窗口的父框架中，打开新的文件。
- 完整窗口：在完成的窗口中打开新的文件。

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 链接。图 11-41 中的设置对应的 HTML 代码如下所示：

```
<a href="index.jsp" target="_top">HTML 链接－返回主页</a>
```

以上代码对应的显示界面如图 11-43 所示。单击该链接会打开设置的 index.jsp 文件。

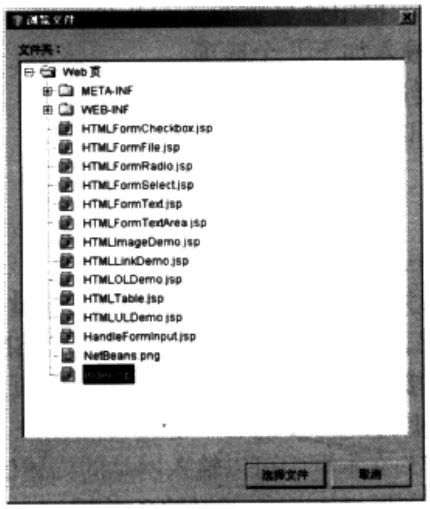


图 11-42 “浏览文件”对话框

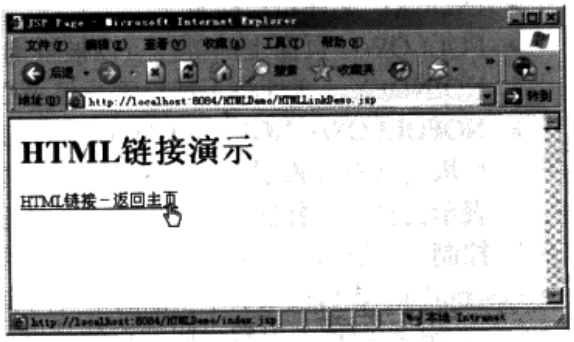


图 11-43 HTML 链接显示界面



11.3.9 使用 HTML 元数据组件

HTML 元数据由 HTML 的<meta>标签组成。在“组件面板”窗口中选中“元数据”组件，

Download at http://www.pin5i.com/

然后拖曳到源代码编辑器中希望出现的地方（通常在<head></head>标签之间）。在释放鼠标后，自动打开“插入 元数据”对话框，如图 11-44 所示。

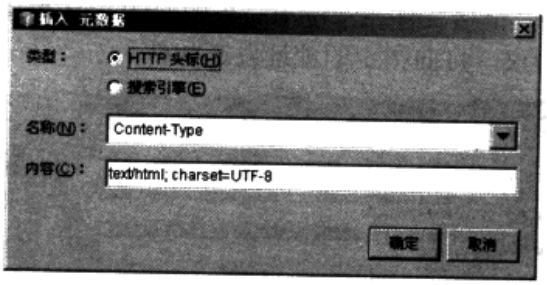


图 11-44 “插入 元数据”对话框

记录当前页面的信息，例如字符编码、作者、版权和关键字等。还可以用来向服务器提供信息，例如截止日期和页面刷新间隔等。它的两种基本用法是：

- 告诉浏览器网页所识别的文件类型及语言类型（如果选择“HTTP 头标”单选按钮）；
- 让搜索引擎按关键字搜索到网页（如果选择“搜索引擎”单选按钮）。

如果选择“HTTP 头标”单选按钮，在“名称”下拉列表框中会显示如下的几个选项。

- Content-Type：用于指定内容的 MIME 类型，在“内容”文本框中可以输入“text/html; charset=UTF-8”，表示使用 UTF-8 编码类型。
- Content-Language：页面使用的文字和语言。在“内容”文本框中可以输入“zh-CN”，表示使用中文。
- Refresh：定时让网页在指定的时间内跳转到页面。在“内容”文本框中可以输入“3;URL=http://www.google.com”，表示在 3 秒内跳转到 google 的主页。
- Cache-Control：表示是否使用缓冲。在“内容”文本框中可以输入“no-cache”表示不使用缓冲。
- Expires：用于设定网页的到期时间，一旦过期则必须到服务器上重新调用。在“内容”文本框中可以输入 GMT 时间格式的时间，例如“星期二，15 四月 2008 18:35:58 CST”。

如果选择“搜索引擎”单选按钮，在“名称”下拉列表框中会显示如下的几个选项。

- robots：设置搜索选项。在“内容”文本框中可以输入如下几个选项。
 - ALL：文件将被检索，且页面上的链接可以被查询。
 - NONE：文件将不被检索，且页面上的链接不可以被查询。
 - INDEX：文件将被检索。
 - FOLLOW：页面上的链接可以被查询。
 - NOINDEX：文件将不被检索，但页面上的链接可以被查询。
 - NOFOLLOW：NOFOLLOW 标签是由 Google 领头新创的一个标签，目的是尽量减少垃圾链接对搜索引擎的影响，减少博客的垃圾留言，目前 Google、Yahoo、MSN 都表示支持这一标签。该标签告诉搜索引擎“该链接所指向的其他网页非当前网页所能控制”。这样，搜索引擎在计算目标的网站的权重时，将会把这个链接排除在外。
- description：告诉搜索引擎站点的主要内容。在“内容”文本框中可以输入网站的描述信息。
- keywords：向搜索引擎说明网页的关键词。在“内容”文本框中可以输入关键词，多个关键词之间使用逗号分隔。

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 元数据。图 11-44 中的设置对应的 HTML 代码如下所示：

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

Download at http://www.pin5i.com/

11.4 在页面中使用 HTML 表单组件

在 HTML 网页上，用户的输入、内容的提交等都是在表单中完成的。本节首先介绍 HTML 表单组件的使用方法，然后介绍表单中的一些组件，例如单行文本输入、多行文本输入、复选框等组件的使用方法。

11.4.1 使用 HTML 表单

HTML 表单由 HTML 的<form></form>标签组成。在“组件面板”窗口中选中“表单”组件，然后拖曳到源代码编辑器中希望出现的地方。在释放鼠标后，自动打开“插入 表单”对话框，如图 11-45 所示。

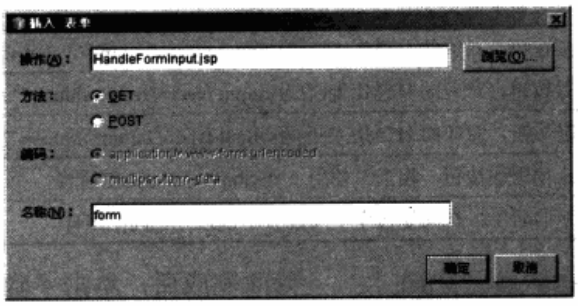


图 11-45 “插入 表单”对话框

在“操作”文本框中可以输入处理表单内容的文件的名字，单击“浏览”按钮可以从项目中选择其他的表单处理文件。

表单的提交有 GET 和 POST 两种方法。当选中 GET 单选按钮，即使用 GET 方法（也是默认的表单提交方法）。当选中 POST 单选按钮后，就选择使用 POST 方法。此时，需要选择 POST 数据的编码类型。

- application/x-www-form-urlencoded 单选按钮：指定了表单数据向服务器提交时所采用的编码类型，默认的属性值是“application/x-www-form-urlencoded”。
- multipart/form-data 单选按钮：在文件上传时，所使用的编码类型应当是使用 enctype="multipart/form-data"属性。它既可以发送文本数据，也支持二进制数据上载。

在“名称”文本框中，可以输入该表单的名字。修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 表单。图 11-45 中的设置对应的 HTML 代码如下所示：

```
<form name="form" action="HandleFormInput.jsp" method="POST">
</form>
```

选中 GET 单选按钮时，对应的 HTML 代码如下所示：

```
<form name="form" action="HandleFormInput.jsp">
</form>
```

11.4.2 使用 HTML 按钮

在 HTML 表单中提交数据或者重置输入的数据时，都需要使用到 HTML 按钮。HTML 按钮由 HTML 的<input type="" value=""/>标签组成。在“组件面板”窗口中选中“按钮”组件，然后拖曳到源代码编辑器中<form></form>标签之间某个希望出现的地方。在释放鼠标后，自动打开“插入 按钮”对话框，如图 11-46 所示。对话框中各个属性的说明如表 11-5 所示。

Download at http://www.pin5i.com/

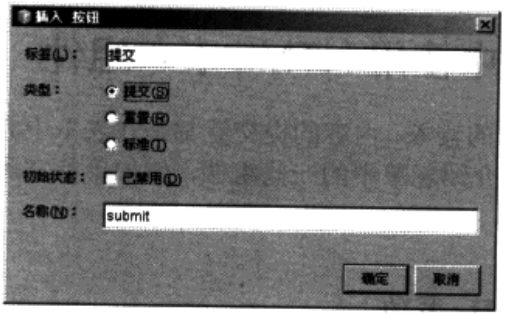


图 11-46 “插入 按钮”对话

表 11-5 HTML 按钮属性

属 性	说 明
标签	按钮上显示的文字
提交	提交按钮，对应的 HTML 标签为<input reset="submit" value="" />
重置	重置按钮，对应的 HTML 标签为<input reset="reset" value="" />
标准	标准按钮，对应的 HTML 标签为<input type="button" value="" />
已禁用	禁止使用该按钮，相当于使用了 disabled="disabled"属性
名称	按钮的名字，在其他程序中通过名字来区分不同的组件

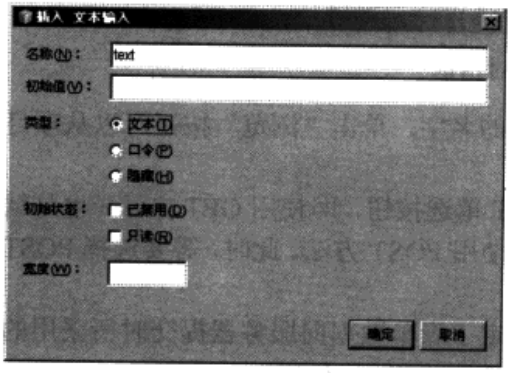


图 11-47 “插入 文本输入”对话框

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 按钮。图 11-46 中的设置对应的 HTML 代码如下所示：

```
    <input type="submit" value="提交"
name="submit" />
```



11.4.3 使用 HTML 文本输入

HTML 文本输入用于输入单行的文本，由 HTML 的<input type="text" name="" value=""/>标签组成。在“组件面板”窗口中选中“文本输入”组件，然后拖曳到源代码编辑器中<form></form>标签之间某个希望出现的地方。释放鼠标后，自动打开“插入 文本输入”对话框，如图 11-47 所示。对话框中各个属性的说明如表 11-6 所示。

表 11-6 HTML 文本输入属性

属 性	说 明
名称	文本输入的名字，在其他程序中通过名字来区分不同的组件
初始值	文本框中的初始文字
文本	标准的文本框，对应的 HTML 标签为<input type="text" name="" value=""/>
口令	用于输入密码的文本框，输入密码时密码显示为其他符号。对应的 HTML 标签为<input type="password" name="" value="" />
隐藏	隐藏的文本框。对应的 HTML 标签为<input type="hidden" name="" value="" />
已禁用	禁止使用该组件，相当于使用了 disabled="disabled"属性
只读	文本框为只读，相当于使用了 readonly="readonly"属性
宽度	文本框的宽度

Download at http://www.pin5i.com/

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 文本输入。图 11-47 中的设置对应的 HTML 代码如下所示：

```
<input type="text" name="text" value="" />
```

在表单 form 中添加一个标准文本框、一个口令类型的文本框和一个隐藏类型的文本框，对应的代码如下所示：

```
<h1>HTML 表单文本输入演示</h1>
<form name="form" action="HandleFormInput.jsp">
    <input type="text" name="text" value="" />
    <p></p>
    <input type="password" name="password" value="123" />
    <p></p>
    <input type="hidden" name="hide" value="hide" />
    <p></p>
    <input type="submit" value="提交" name="submit" />
</form>
```

以上代码对应的显示界面如图 11-48 所示。

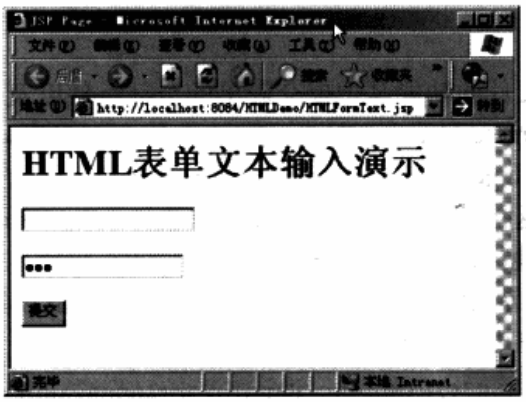


图 11-48 HTML 文本输入显示界面

这里，表单使用默认的 GET 方法提交数据，数据被提交到 HandleFormInput.jsp 文件进行处理。HandleFormInput.jsp 文件的具体内容如下所示：

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<%@ page import="java.util.Enumeration" %>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>处理表单输入</h1>
        <p></p>

        <table border="1">
            <thead>
                <tr>
                    <th>参数名字</th>
                    <th>参数值</th>
                </tr>
            </thead>
            <tbody>
                <%
                Enumeration paramNames = request.getParameterNames();
                while(paramNames.hasMoreElements()) {
```

Download at http://www.pin5i.com/


```
                String paramName = (String)paramNames.nextElement();
                out.print("<tr><td>"+paramName+"</td><td>");
                String[] paramValues = request.getParameterValues(paramName);
                if (paramValues.length == 1) {
                    String paramValue = paramValues[0];
                    if (paramValue.length() == 0) {
                        out.print("<I>没有值</I>");
                    } else {
                        out.print(paramValue);
                    }
                } else {
                    out.println("<ul>");
                    for(int i=0; i<paramValues.length; i++) {
                        out.println("<li>" + paramValues[i]);
                    }
                    out.println("</ul>");
                }
                out.println("</td></tr>");
            }
            %>
        </tbody>
    </table>
</body>
</html>
```

HandleFormInput.jsp 文件的功能是把提交表单的全部组件以及组件值以表格的形式打印出来。其中，request.getParameterNames()方法返回请求的全部参数，这里对应的是提交表单中 HTML 组件的名字。request.getParameterValues(paramName)方法返回指定参数的参数值，这里给出表单中组件的名字，返回组件的值。out.print()方法和 out.println()方法用于在页面上打印输出指定的内容，这里巧妙地控制输出一个表格。

在如图 11-48 所示的界面中，单击“提交”按钮，表单的内容被以 URL 的形式提交到 HandleFormInput.jsp 文件中进行处理。提交的 URL 为：

```
http://localhost:8084/HTMLDemo/HandleFormInput.jsp?text=&password=123&hide=h
ide&submit=%E6%8F%90%E4%BA%A4
```

HandleFormInput 文件会显示表单的全部内容，如图 11-49 所示。



图 11-49 处理文本输入组件的输入值



11.4.4 使用 HTML 多行输入

HTML 多行输入用于输入多行的文本，由 HTML 的<textarea name="textarea" rows="" cols="">标签组成。在“组件面板”窗口中选中“多行输入”组件，然后拖曳到源代码编辑器中<form></form>标签之间某个希望出现的地方。释放鼠标后，自动打开“插入 多行输入”对话框，如图 11-50 所示。对话框中各个属性的说明如表 11-7 所示。

Download at http://www.pin5i.com/

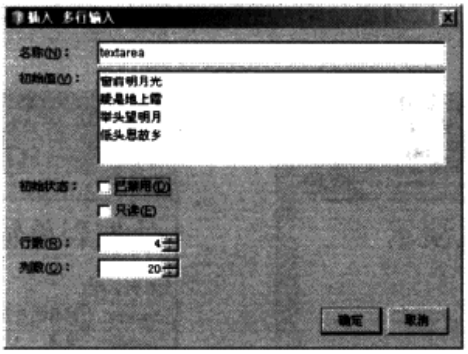


图 11-50 “插入 多行输入”对话框

表 11-7 HTML 多行输入属性

属 性	说 明
名称	多行输入的名字，在其他程序中通过名字来区分不同的组件
初始值	文本框中的初始文字，可以是多行的文字
已禁用	禁止使用该组件，相当于使用了 disabled="disabled"属性
只读	文本框为只读，相当于使用了 readonly="readonly"属性
行数	多行输入的行数限制
列数	多行输入的列数限制

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 多行输入。图 11-50 中的设置对应的 HTML 代码如下所示：

```
<textarea name="textarea" rows="4" cols="20">
窗前明月光
疑是地上霜
举头望明月
低头思故乡
</textarea>
```

在表单 form 中添加一个多行输入和一个提交按钮，对应的代码如下所示：

```
<h1>HTML 表单多行输入演示</h1>
<form name="form" action="HandleFormInput.jsp">
   <textarea name="textarea" rows="4" cols="20">
窗前明月光
疑是地上霜
举头望明月
低头思故乡
   </textarea>
   <p></p>
   <input type="submit" value="提交" name="submit" />
</form>
```

以上代码对应的显示界面如图 11-51 所示。

单击“提交”按钮后，HandleFormInput.jsp 文件会显示表单中的组件以及组件值，如图 11-52 所示。



11.4.5 使用 HTML 下拉列表

HTML 下拉列表由 HTML 的<select></select>标签组成。在“组件面板”窗口中选中“多行输入”组件，然后拖曳到源代码编辑器中<form></form>标签之间某个希望出现的地方。释

Download at http://www.pin5i.com/

放鼠标后，自动打开“插入 下拉列表”对话框，如图 11-53 所示。对话框中各个属性的说明如表 11-8 所示。



图 11-51 HTML 多行输入显示界面



图 11-52 处理多行输入组件的输入值

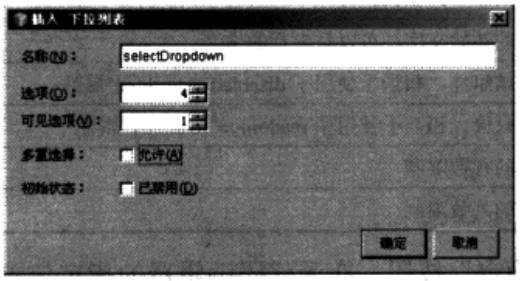


图 11-53 “插入 下拉列表”对话框

表 11-8 HTML 下拉列表属性

属 性	说 明
名称	下拉列表的名字，在其他程序中通过名字来区分不同的组件
选项	下拉列表的选项个数
可见选项	可见到选项个数
多重选择：允许	是否可以同时选中多个选项，如果未选中该选项，则显示形式为下拉列表框；否则显示为列表框形式
已禁用	禁止使用该组件，相当于使用了 disabled="disabled"属性

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 下拉列表。图 11-53 中的设置对应的 HTML 代码如下所示：

```
<select name="selectDropDown">
    <option></option>
    <option></option>
    <option></option>
    <option></option>
</select>
```

在表单 form 中添加两个下拉列表和一个提交按钮，对应的代码如下所示：

```
<h1>HTML 表单下拉列表演示</h1>
<form name="form" action="HandleFormInput.jsp">
    <select name="selectDropDown">
        <option>窗前明月光</option>
        <option>疑是地上霜</option>
        <option>举头望明月</option>
        <option>低头思故乡</option>
```

Download at http://www.pin5i.com/


```
    </select>
    <p></p>
    <select name="selectList" size="4" multiple="multiple">
        <option>窗前明月光</option>
        <option>疑是地上霜</option>
        <option>举头望明月</option>
        <option>低头思故乡</option>
    </select>
    <p></p>
    <input type="submit" value="提交" name="submit" />
</form>
```

以上代码对应的显示界面如图 11-54 所示。

单击“提交”按钮后，HandleFormInput.jsp 文件会显示表单中的组件以及组件值，如图 11-55 所示。



图 11-54 HTML 下拉列表显示界面



图 11-55 处理下拉列表组件的输入值



11.4.6 使用 HTML 复选框

HTML 复选框由 HTML 的<input type="checkbox" name="" value="" />标签组成。在“组件面板”窗口中选中“复选框”组件，然后拖曳到源代码编辑器中<form></form>标签之间某个希望出现的地方。释放鼠标后，自动打开“插入 复选框”对话框，如图 11-56 所示。对话框中各个属性的说明如表 11-9 所示。

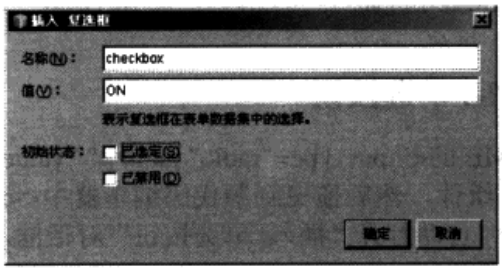


图 11-56 “插入 复选框”对话框

表 11-9 HTML 复选框属性

属 性	说 明
名称	复选框的名字，在其他程序中通过名字来区分不同的组件
值	选中该复选框对应的值
已选定	复选框是否被选中。选中该选项后添加 checked="checked"属性
已禁用	禁止使用该组件，相当于使用了 disabled="disabled"属性

Download at http://www.pin5i.com/

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 复选框。图 11-56 中的设置对应的 HTML 代码如下所示：

```
<input type="checkbox" name="checkbox" value="ON" />
```

在表单 form 中添加四个复选框和一个提交按钮，对应的代码如下所示：

```
<h1>HTML 表单复选框演示</h1>
<form name="form" action="HandleFormInput.jsp">
    <input type="checkbox" name="checkbox" value="窗前明月光"
        checked="checked" />窗前明月光
    <br>
    <input type="checkbox" name="checkbox"
        value="疑是地上霜" />疑是地上霜
    <br>
    <input type="checkbox" name="checkbox"
        value="举头望明月" />举头望明月
    <br>
    <input type="checkbox" name="checkbox"
        value="低头思故乡" />低头思故乡
    <p></p>
    <input type="submit" value="提交" name="submit" />
</form>
```

其中，这几个复选框的名字都为 checkbox，则可以把这些复选框作为一个组件进行处理（组件值有多个）。以上代码对应的显示界面如图 11-57 所示。

单击“提交”按钮后，HandleFormInput.jsp 文件会显示表单中的组件以及组件值，如图 11-58 所示。

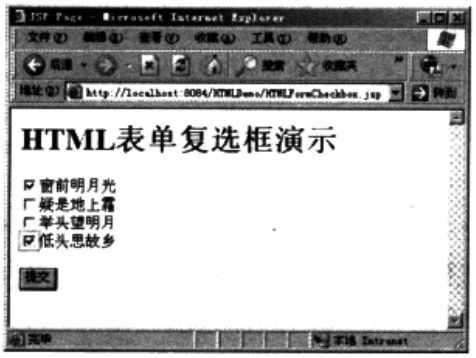


图 11-57 HTML 复选框显示界面

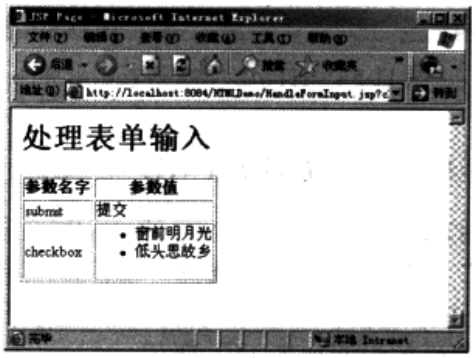


图 11-58 处理复选框组件的输入值



11.4.7 使用 HTML 单选按钮

HTML 单选按钮由 HTML 的<input type="radio" name="" value="" />标签组成。在“组件面板”窗口中选中“复选框”组件，然后拖曳到源代码编辑器中<form></form>标签之间某个希望出现的地方。释放鼠标后，自动打开“插入 单选按钮”对话框，如图 11-59 所示。对话框中各个属性的说明如表 11-10 所示。

表 11-10 HTML 单选按钮属性

属 性	说 明
组	单选按钮组的名字，如果几个单选按钮使用相同的组名字，那么选中组中的某个单选按钮时，其他单选按钮就变为未被选中状态。即在一个组中的单选按钮，同一时间只能有一个单选按钮被选中
值	选中该单选按钮对应的值
已选定	单选按钮是否被选中。选中该选项后添加 checked="checked"属性
已禁用	禁止使用该组件，相当于使用了 disabled="disabled"属性

Download at http://www.pin5i.com/

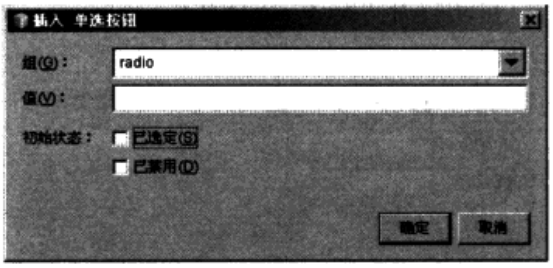


图 11-59 “插入 单选按钮”对话框

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 单选按钮。图 11-59 中的设置对应的 HTML 代码如下所示：

```
<input type="radio" name="radio" value="" />
```

在表单 form 中添加四个单选按钮和一个提交按钮，对应的代码如下所示：

```
<h1>HTML 表单单选按钮演示</h1>
<form name="form" action="HandleFormInput.jsp">
    <input type="radio" name="radio"
          value="窗前明月光" />窗前明月光
    <br>
    <input type="radio" name="radio"
          value="疑是地上霜" />疑是地上霜
    <br>
    <input type="radio" name="radio"
          value="举头望明月" />举头望明月
    <br>
    <input type="radio" name="radio"
          value="低头思故乡" checked="checked" />低头思故乡
    <p></p>
    <input type="submit" value="提交" name="submit" />
</form>
```

其中，这几个单选按钮的名字都为 radio。以上代码对应的显示界面如图 11-60 所示。

单击“提交”按钮后，HandleFormInput.jsp 文件会显示表单中的组件以及组件值，如图 11-61 所示。



图 11-60 HTML 单选按钮显示界面



图 11-61 处理单选按钮组件的输入值



11.4.8 使用 HTML 选择文件

HTML 选择文件由 HTML 的<input type="file" name="" value="" />标签组成。在“组件面板”窗口中选中“复选框”组件，然后拖曳到源代码编辑器中<form></form>标签之间某个希

Download at http://www.pin5i.com/

望出现的地方。释放鼠标后，自动打开“插入 选择文件”对话框，如图 11-62 所示。对话框中各个属性的说明如表 11-11 所示。

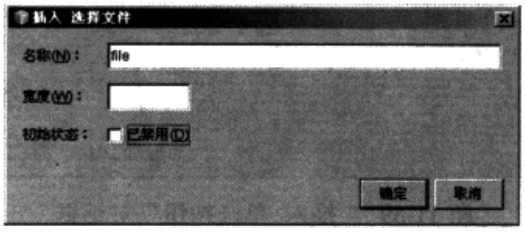


图 11-62 “插入 选择文件”对话框

表 11-11 HTML 选择文件属性

属 性	说 明
名称	选择文件的名字，在其他程序中通过名字来区分不同的组件
宽度	选择文件组件的宽度
已禁用	禁止使用该组件，相当于使用了 disabled="disabled"属性

修改完成后，单击“确定”按钮，在源代码编辑器的指定位置会自动地插入 HTML 选择文件。图 11-62 中的设置对应的 HTML 代码如下所示：

```
<input type="file" name="file" value="" />
```

在表单 form 中添加一个选择文件和一个提交按钮，对应的代码如下所示：

```
<h1>HTML 表单选择文件演示</h1>
<form name="form" action="HandleFormInput.jsp">
    <input type="file" name="file" value="" />
    <p></p>
    <input type="submit" value="提交" name="submit" />
</form>
```

以上代码对应的显示界面如图 11-63 所示。

单击“提交”按钮后，HandleFormInput.jsp 文件会显示表单中的组件以及组件值，如图 11-64 所示。



图 11-63 HTML 选择文件显示界面



图 11-64 处理选择文件组件的输入值

11.5 基于 JSP 的开发实例——登录模块的实现

在前两节中已经介绍了如何在 NetBeans 中使用 HTML 组件。本节将在此基础上，利用 JSP 技术开发一个登录模块，对 JSP Web 应用程序的开发进行一个综合的演示。

Download at http://www.pin5i.com/

11.5.1 登录模块实现流程

登录模块是比较常用的一个模块，各个系统一般都需要用户首先进行登录，然后才能够使用一些功能。简单的登录模块一般需要四个基本的文件，其实现流程如图 11-65 所示。

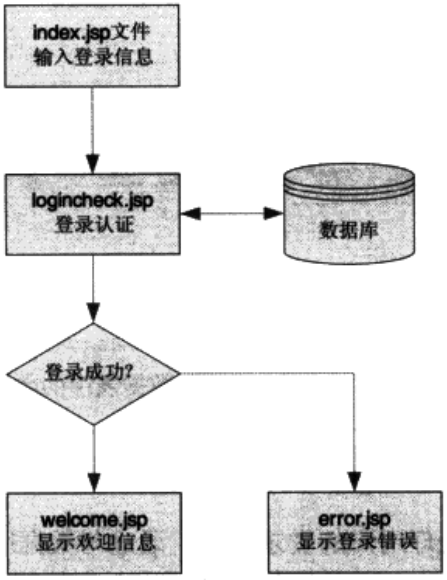


图 11-65 登录模块实现流程

用户首先看到的是 index.jsp 文件，并在该文件中输入用户名及密码。在提交输入后，logincheck.jsp 文件会根据用户的输入，在数据库中查找用户信息，并对用户的密码进行验证。如果验证通过，则打开 welcome.jsp 文件，显示欢迎信息，否则打开 error.jsp 文件，显示登录错误信息。

11.5.2 设计并创建数据库

用户的账号信息以及登录密码一般都保存在数据库中。在本实例中采用 Java DB 数据库，按照第 7 章中介绍的方法，创建一个名为 MyTest 的 Java DB 数据库。数据库的用户名为 admin，密码为 111。

运行下面的代码，创建一个名为 USERS 的数据表，保存用户的账号信息。

```
create table "ADMIN"."USERS"
(
 "UserID" INTEGER not null unique,
 "UserName" VARCHAR(12) not null unique,
 "Password" VARCHAR(20)
)
```

创建表后，执行如下的 SQL 语句，为表插入一条记录。即添加用户 Robin，密码为 111。

```
insert into "USERS"
("UserID","UserName","Password")
values (1,'Robin','111');
```

当执行如下的代码时，如果显示如图 11-66 所示的界面，则表示数据库已经准备好了。

```
select * from "ADMIN"."USERS"
```

Download at http://www.pin5i.com/

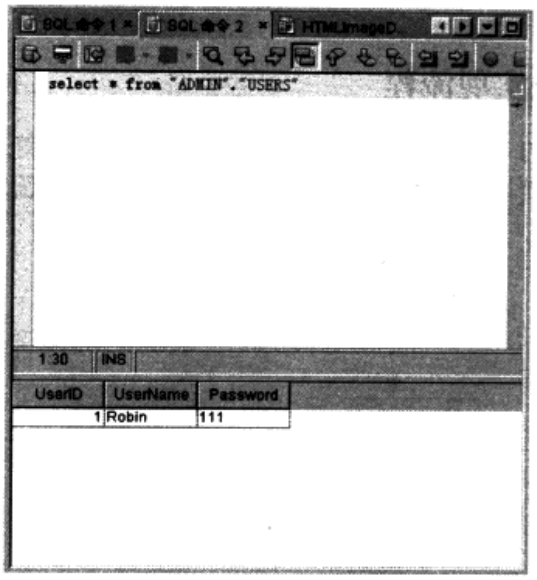


图 11-66 查询 USERS 数据表中的数据



11.5.3 创建 logincheck.jsp 文件

logincheck.jsp 文件用于处理用户的登录信息，通过与数据库中 USERS 表中的数据进行比较，对用户登录的请求进行检验。下面是 logincheck.jsp 文件创建的具体过程。

按照第 11.2.1 节中介绍的方法，创建一个 Web 应用程序项目，名字为 JSP_Login。服务器使用捆绑的 Tomcat 服务器，把该项目设置为主项目。

按照第 11.3.2 节中介绍的方法，创建一个使用标准 JSP 语法的 JSP 页面，名字为 logincheck.jsp。将文件的具体内容修改为如下所示：

```
<%@ page contentType="text/html; charset=GBK" %>
<%@page import="java.sql.*"%>
<%
String username = request.getParameter("username");
String password = request.getParameter("password");
boolean loginsuccess = false;


Connection con = null;
Statement stmt = null;
ResultSet rs = null;

try {
    Class.forName("org.apache.derby.jdbc.ClientDriver");
    con = DriverManager.getConnection("jdbc:derby://localhost:1527/MyTest",
          "admin", "111");
    stmt = con.createStatement();

    String sql = "select * from USERS where \"UserName\" = '" + username + "'";
    rs = stmt.executeQuery(sql);
    while (rs.next()) {
        if(password.equals(rs.getString("Password"))){
            loginsuccess=true;
            session.setAttribute("username",username);
        }
    }

} catch (Exception sqlex1) {
    sqlex1.printStackTrace();
} finally {
```

Download at http://www.pin5i.com/


```
        if (con != null) {
            try {
                con.close();
            } catch (SQLException sqlex2) {
                sqlex2.printStackTrace();
            }
        }

    }

    if (loginsuccess) {
    %>
    <jsp:forward page="welcome.jsp"></jsp:forward>
    <%
    } else {
    %>
    <jsp:forward page="error.jsp"></jsp:forward>
    <%
    }
    %>
```

其中，下面的两行语句用于获取用户输入的用户名以及密码：

```
String username = request.getParameter("username");
String password = request.getParameter("password");
```

下面的语句用于从数据库 USERS 表中查找用户，如果找到则比较密码是否与输入的一致。如果一致则认为登录成功，把登录标志 loginsuccess 设置为 true，并把用户名保存在会话中。

```
Class.forName("org.apache.derby.jdbc.ClientDriver");
con = DriverManager.getConnection("jdbc:derby://localhost:1527/MyTest",
        "admin", "111");
stmt = con.createStatement();

String sql = "select * from USERS where \"UserName\" = '" + username + "'";
rs = stmt.executeQuery(sql);
while (rs.next()) {
    if(password.equals(rs.getString("Password"))){
        loginsuccess=true;
        session.setAttribute("username",username);
    }
}
```

Java DB 数据库的驱动程序名字以及 MyTest 数据库的连接字符串可以通过以下的方式获得。打开“运行环境”窗口，右键单击 MyTest 数据库，选择弹出菜单中的“连接”命令，连接到数据库。再次单击 MyTest 数据库，选择弹出菜单中的“属性”命令，打开数据库的属性对话框，如图 11-67 所示。“数据库 URL”属性就是数据库的连接字符串；“驱动程序”属性就是 Class.forName()方法中使用的参数。

如果登录标记 loginsuccess 为 true，那么就通过 JSP 标记<jsp:forward></jsp:forward>打开欢迎页面；否则通过 JSP 标记<jsp:forward></jsp:forward>打开错误页面。

```
if (loginsuccess) {
%>
<jsp:forward page="welcome.jsp"></jsp:forward>
<%
} else {
%>
<jsp:forward page="error.jsp"></jsp:forward>
<%
}
%>
```

Download at http://www.pin5i.com/

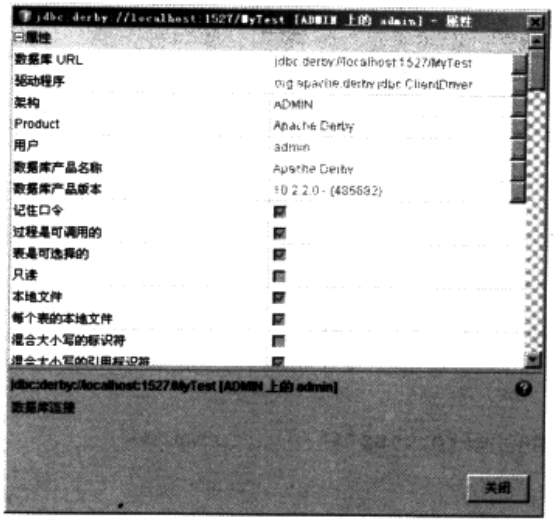


图 11-67 数据库的属性对话框



11.5.4 创建 index.jsp 文件

index.jsp 文件提供了用户输入账号以及密码的界面，也是用户首先看到的界面。下面是 index.jsp 文件创建的具体过程。

❶ 打开创建 Web 应用程序时创建的 index.jsp 文件。从“组件面板”窗口中拖曳一个“表单”组件，打开“插入 表单”对话框，如图 11-68 所示。

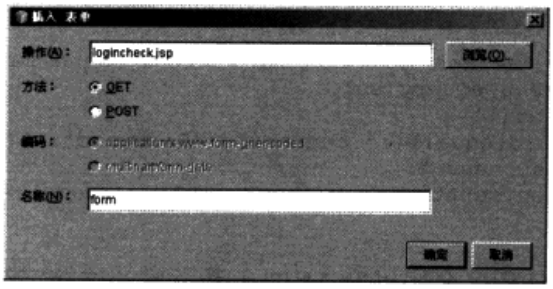


图 11-68 “插入 表单”对话框

❷ 在“名称”文本框中输入表单名“form”，单击“浏览”按钮，打开“浏览文件”对话框，如图 11-69 所示。

❸ 选中 logincheck.jsp 文件，单击“选择文件”按钮，返回“插入 表单”对话框，再单击“确定”按钮，在源代码中插入表单。

❹ 在“组件面板”窗口中拖曳一个“文本输入”组件，打开“插入 文本输入”对话框，如图 11-70 所示。

❺ 在“名称”文本框中输入“username”，这里要确保输入的名字应该和 logincheck.jsp 文件中 request.getParameter("username")的参数名字一致。在“初始值”文本框中输入“Robin”，作为文本框中的默认文字。单击“确定”按钮，在源代码中插入文本输入。

❻ 在“组件面板”窗口中再拖曳一个“文本输入”组件，打开“插入 文本输入”对话框，如图 11-71 所示。

❼ 在“名称”文本框中输入“password”，这里要确保输入的名字应该和 logincheck.jsp 文件中 request.getParameter("password")的参数名字一致。选中“口令”单选按钮，这样输入口令时输入的内容将被其他字符代替。单击“确定”按钮，在源代码中插入文本输入。

Download at http://www.pin5i.com/

❽ “组件面板”窗口中拖曳一个“按钮”组件，打开“插入 按钮”对话框，如图 11-72 所示。在“标签”文本框中输入按钮上显示的文字“确认”，在“名称”文本框中输入按钮的名字“submit”。单击“确定”按钮，在源代码中插入按钮。

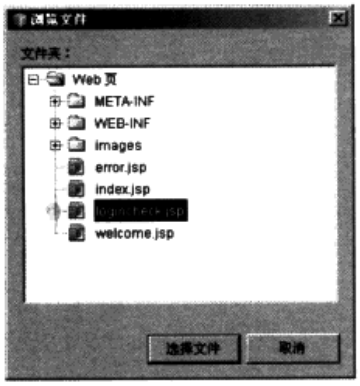


图 11-69 “浏览文件”对话框

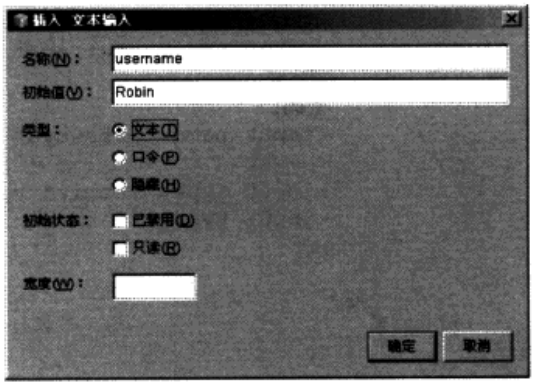


图 11-70 “插入 文本输入”对话框

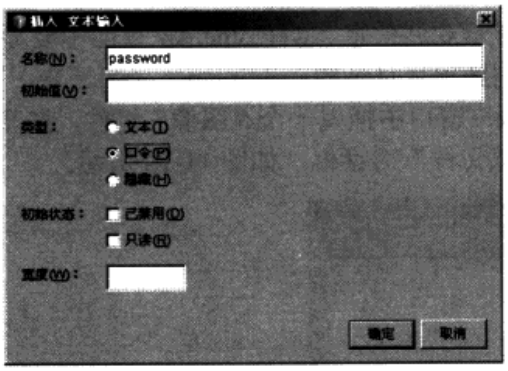


图 11-71 “插入 文本输入”对话框

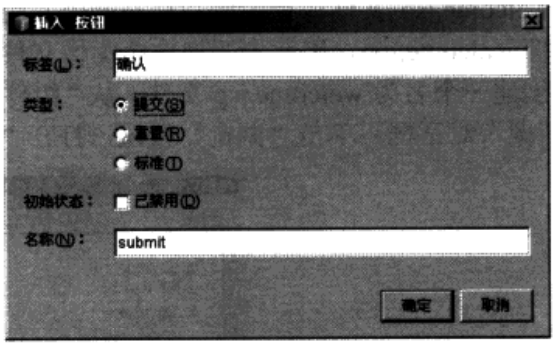


图 11-72 “插入 按钮”对话框 1

❾ “组件面板”窗口中再拖曳一个“按钮”组件，打开“插入 按钮”对话框，如图 11-73 所示。选中“重置”单选按钮，在“名称”文本框中输入按钮的名字“reset”。单击“确定”按钮，在源代码中插入按钮。

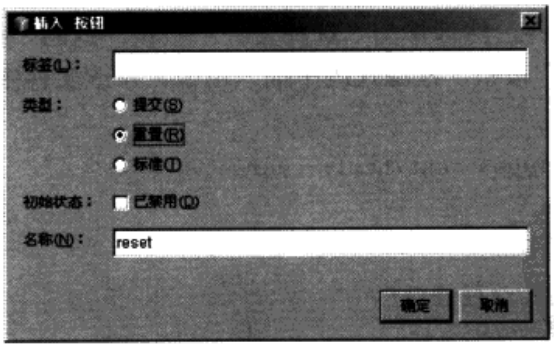


图 11-73 “插入 按钮”对话框 2

❿ 对 index.jsp 文件的代码进行修改，最后的 index.jsp 文件内容如下所示。

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP login</title>
```

Download at http://www.pin5i.com/


```
    </head>
    <body>

        <h1>请登录系统</h1>
        <form action="logincheck.jsp" method="post">
            用户:
            <input name="username" value="Robin"/>
            <br>
            密码:
            <input name="password" type="password" value=""/>
            <p></p>
            <input type="submit" value="确认"/>
            <input type="reset"/>
        </form>

    </body>
</html>
```



11.5.5 创建 welcome.jsp 文件

welcome.jsp 文件在登录成功后被调用，用于显示欢迎信息。welcome.jsp 页面中要显示一个图像 welcome.jpg，因此先要把这个图像文件复制到项目目录的 web\images 子目录中。

创建一个名为 welcome.jsp 文件。从“组件面板”窗口中拖曳一个“图像”组件，打开“插入 图像”对话框。单击“浏览”按钮，打开“浏览文件”对话框，如图 11-74 所示。

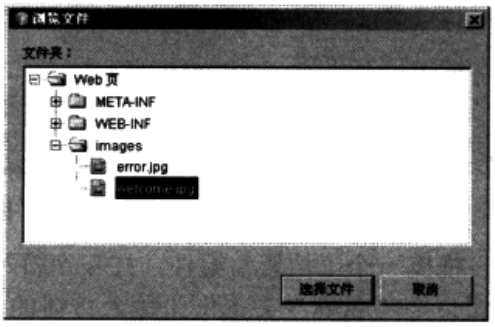


图 11-74 “浏览文件”对话框

选择 welcome.jpg 文件，然后单击“选择文件”按钮，返回“插入 图像”对话框，如图 11-75 所示。单击“确定”按钮，在源代码中插入图像。修改文件中的其他代码，最终代码如下所示：

```
<%@page contentType="text/html; charset=GBK"%>
<html>
    <head>
        <title>welcome</title>
    </head>
    <body bgcolor="#ffffff">
        <h1>登录成功</h1>
        欢迎您:
        <%out.println(session.getAttribute("username"));%>
        <p></p>
        <img src="images/welcome.jpg" width="555" height="225" alt="welcome"/>

    </body>
</html>
```

其中，out.println 用于在页面上打印输出信息。session.getAttribute("username")用于读取会话中保存的登录用户的名字。

Download at http://www.pin5i.com/

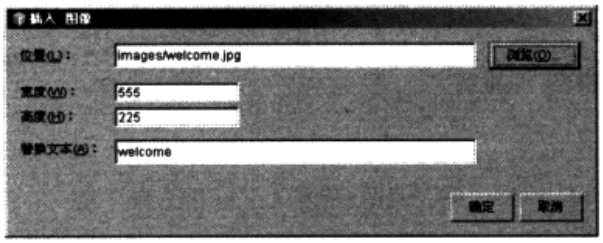


图 11-75 “插入 图像”对话框

11.5.6 创建 error.jsp 文件

welcome.jsp 文件在登录成功后被调用，用于显示欢迎信息。error.jsp 页面中要显示一个图像 error.jpg，因此先要把这个图像文件复制到项目目录的 web\images 子目录中。按照第 11.5.5 节中的方法创建 error.jsp 文件，并添加 HTML 图像组件。

从组件面板”窗口中拖曳一个“链接”组件，打开“插入 链接”对话框。单击“浏览”按钮，打开“浏览文件”对话框，如图 11-76 所示。这里希望用户单击该链接后返回用户登录界面，因此要选中 index.jsp 文件。

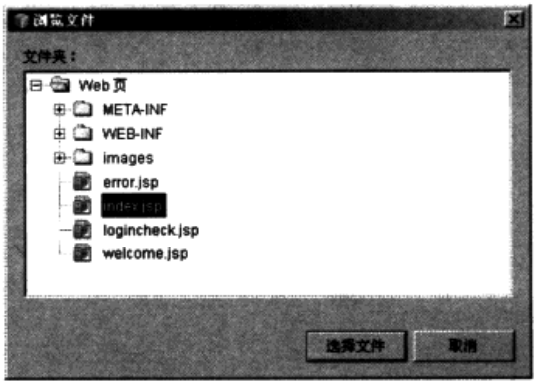


图 11-76 “浏览文件”对话框

单击“选择文件”按钮，返回“插入 链接”对话框，如图 11-77 所示。在“文本”文本框中输入“返回”，单击“确定”按钮，在源代码中插入链接。

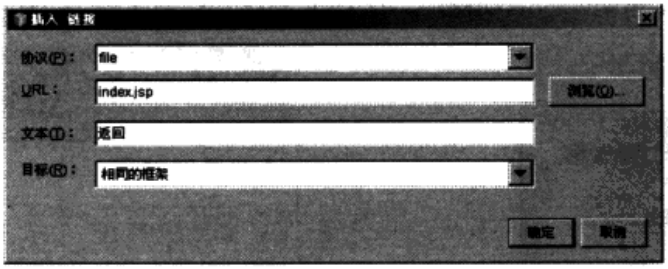


图 11-77 “插入 链接”对话框

修改文件中的其他代码，最终代码如下所示：

```
<%@ page contentType="text/html; charset=GBK" %>
<html>
    <head>
        <title>
            error
        </title>
    </head>
```

Download at http://www.pin5i.com/


```
        <body bgcolor="#ffffff">
            <h1>
                <img src="images/error.jpg" width="191" height="199" alt="error"/>
                登录错误，请重试
            </h1>
            <p></p>
            <p></p>
            <a href="index.jsp">返回</a>
        </body>
    </html>
```



11.5.7 运行项目

在运行项目前，有两个事情需要首先处理。首先要确认 MyTest 数据库已经启动，其次要在“项目”窗口中右键单击 JSP_Login 项目，选择弹出菜单中的“属性”命令，打开“项目属性”对话框，再单击“库”节点，显示如图 11-78 所示的界面。

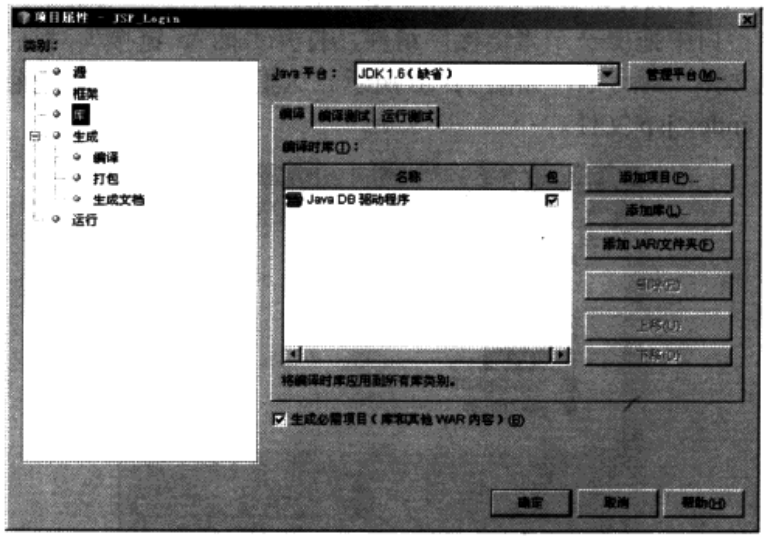


图 11-78 “项目属性”对话框的“库”节点

单击“添加库”按钮，打开“添加库”对话框，如图 11-79 所示。在“库”列表中选择“Java DB 驱动程序”节点，然后单击“添加库”按钮，返回“项目属性”对话框。单击“确定”按钮，在项目中添加 Java DB 数据库的驱动程序。运行主项目，显示如图 11-80 所示的用户登录界面。

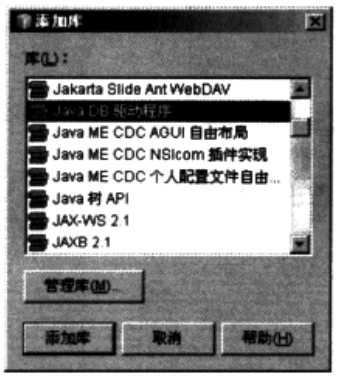


图 11-79 “添加库”对话框

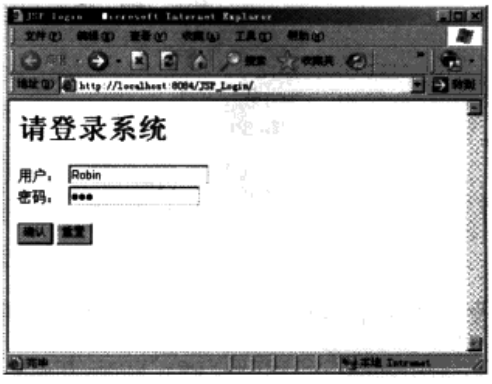


图 11-80 用户登录界面

在输入“用户”和“密码”后，单击“确认”按钮。如果密码输入正确，会显示如图 11-81 所示的欢迎界面，否则会显示如图 11-82 所示的错误信息界面。

Download at http://www.pin5i.com/



图 11-81 欢迎界面



图 11-82 错误信息界面

11.6 在 JSP 页面中使用 JavaBean

在 JSP 页面中，可以使用 JavaBean 来封装事务逻辑、数据库操作等，实现业务逻辑和前台程序的分离。事实上，JSP+JavaBean 是非常常见的一种开发模式。本节首先介绍一下在 JSP 中使用 JavaBean 的基本方法，然后通过一个登录模块的实例进行具体说明。

11.6.1 在 JSP 页面中声明 JavaBean

在 JSP 页面中，使用 JavaBean 前首先要声明 JavaBean。JavaBean 的声明通过<jsp:useBean>动作来实现，具体语法定义如下所示：

```
<jsp:useBean id="" class="" scope=""/>
```

其中，<jsp:useBean>是 JSP 的标准动作，用于创建指定的 class 类。class 属性用于指定要创建的 JavaBean 类。id 属性是创建 JavaBean 实例的名字，下面的代码中都通过 id 定义的名字来使用 JavaBean 实例。属性 scope 指定了 JavaBean 的作用范围，可以有如下的几个属性值。

- application：在整个应用程序范围内有效。
- page：在当前页面范围内有效。
- request：在当前请求范围内有效。
- session：在当前会话范围内有效。

在“组件面板”窗口中，选择“使用 Bean”组件，然后拖曳到源代码编辑器中。释放鼠标后，会显示如图 11-83 所示的“插入 使用 Bean”对话框。

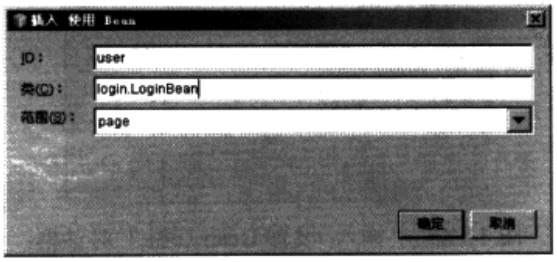


图 11-83 “插入 使用 Bean”对话框

Download at http://www.pin5i.com/

在“ID”文本框中输入 Bean 的名字；在“类”文本框中给出 Bean 类。在“范围”下拉列表框中可以选择 Bean 的作用范围。单击“确定”按钮，声明了一个 login.LoginBean 实例，实例的名字为 user。JavaBean 的有效范围是在当前页面中。

```
<jsp:useBean id="user" class="login.LoginBean" scope="page">
</jsp:useBean>
```



11.6.2 读取 Bean 属性

JavaBean 的属性可以通过<jsp:getProperty>动作获取，语法定义如下所示：

```
<jsp:getProperty name="" property=""/>
```

其中，name 属性用于指定要读取的 Bean 的名字。property 属性为要读取的属性名字。在“组件面板”窗口中，选择“获取 Bean 属性”组件，然后拖曳到源代码编辑器中。释放鼠标后，会显示如图 11-84 所示的“插入 获取 Bean 属性”对话框。

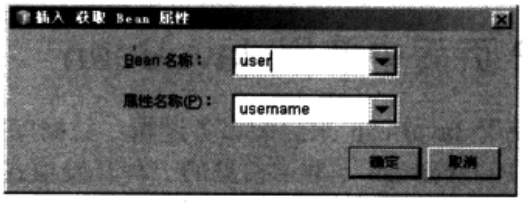


图 11-84 “插入 获取 Bean 属性”对话框

在“Bean 名称”下拉列表框中输入要使用的 Bean 名字，或者通过下拉列表框选择系统对象。在“属性名称”文本框中输入要读取的属性名字。单击“确定”按钮，创建读取属性的代码，如下所示：

```
<jsp:getProperty name="user" property="username" />
```



11.6.3 设置 Bean 属性

JavaBean 的属性可以通过<jsp:setProperty>动作设置，语法定义如下所示：

```
<jsp:setProperty name="" property="*"/>
```

其中，name 属性为要设置的 JavaBean 的名字。property 属性用于指定哪个属性要被设置，如果属性值为“*”，那么会使用提交表单中的所有值来填充 JavaBean 的属性。在“组件面板”窗口中，选择“设置 Bean 属性”组件，然后拖曳到源代码编辑器中。释放鼠标后，会显示如图 11-85 所示的“插入 设置 Bean 属性”对话框。

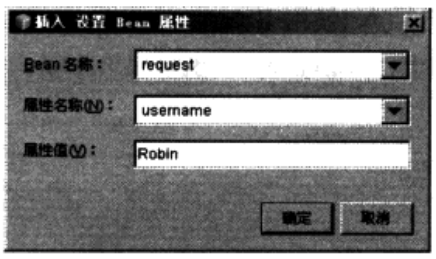


图 11-85 “插入 设置 Bean 属性”对话框

在“Bean 名称”下拉列表框中输入要使用的 Bean 名字，或者通过下拉列表框选择系统对象。在“属性名称”文本框中输入要设置的属性名字。在“属性值”文本框中输入希望设置的

Download at http://www.pin5i.com/

属性值。单击“确定”按钮，创建设置属性的代码，如下所示：

```
<jsp:setProperty name="user" property="username" value="Robin" />
```

下面的代码用于创建一个 LoginBean 的 Bean，然后使用提交的表单数据初始化 Bean 的属性值。

```
<jsp:useBean id="user" class="login.LoginBean" scope="page">
   <jsp:setProperty name="user" property="*"/>
</jsp:useBean>
```

在 JavaBean 创建以后，就可以在下面的代码中像使用一个对象一样使用 JavaBean 了。下面的代码就调用 JavaBean 的 checklogin()方法，并将返回值保存在 loginseccess 变量中。

```
<%
boolean loginseccess = false;
loginseccess = user.checklogin();
%>
```



11.6.4 基于 JSP + JavaBean 的开发实例——登录模块的实现

本节将对第 11.5 节中创建的登录模块代码进行修改，把对数据库的操作封装在 JavaBean 中，然后在 JSP 页面中调用 JavaBean 进行登录校验。

❶ 创建一个 Web 应用程序，名字为“JSP_JavaBean_Login”，并设置为主项目。

❷ 把第 11.5 节中创建的 JSP 文件以及图像全部复制到当前项目目录的 web 子目录中。

❸ 创建一个 JavaBean，名字为 LoginBean，代码如下所示。在 LoginBean 中定义了 username 和 password 两个属性，用于保存表单中输入的用户名和密码。checklogin 方法用于读取数据库，进行登录检验。该方法中的大部分代码来自于 logincheck.jsp，并进行了必要的修改。

```
package login;
import java.sql.*;

public class LoginBean {
   private String username;
   private String password;

   public void setUsername(String _username) {
      this.username = _username;
   }

   public String getUsername() {
      return this.username;
   }

   public void setPassword(String _password) {
      this.password = _password;
   }

   public String getPassword() {
      return this.password;
   }

   public boolean checklogin() {
      boolean loginflag = false;


      Connection con = null;
      Statement stmt = null;
      ResultSet rs = null;

      try {
         Class.forName("org.apache.derby.jdbc.ClientDriver");
         con =
```

Download at http://www.pin5i.com/


```
                DriverManager.getConnection("jdbc:derby://localhost: 1527/MyTest",
                        "admin", "111");
                stmt = con.createStatement();

                String sql = "select * from USERS where \"UserName\" = '" + username
    + "'";
                rs = stmt.executeQuery(sql);
                while (rs.next()) {
                    if(password.equals(rs.getString("Password"))){
                        loginflag=true;
                    }
                }
            } catch (Exception sqlex1) {
                sqlex1.printStackTrace();
            } finally {
                if (con != null) {
                    try {
                        con.close();
                    } catch (SQLException sqlex2) {
                        sqlex2.printStackTrace();
                    }
                }
            }

            return loginflag;
        }
    }
```

❹ 把 logincheck.jsp 文件进行修改，来使用 LoginBean 进行登录检验。这样可以实现业务逻辑和显示页面的分离，提高了代码的可读行以及可扩展性。

```
<%@ page contentType="text/html; charset=GBK" %>

<%@page import="login.LoginBean"%>
<jsp:useBean id="user" class="login.LoginBean" scope="page">
    <jsp:setProperty name="user" property="*"/>
</jsp:useBean>

<%
if (user.checklogin()) {
    String username = request.getParameter("username");
    session.setAttribute("username", username);
%>
<jsp:forward page="welcome.jsp"></jsp:forward>
<%} else {%>
<jsp:forward page="error.jsp"></jsp:forward>
<%}%>
```

❺ 按照第 11.5.7 节中介绍的方法运行程序。程序的运行结果如第 11.5.7 节中的界面所示。

11.7 小结

本章主要简单 Web 应用程序的开发，即简单 JSP 应用程序的创建、HTML/XHTML/JSP 页面的编辑、如何在 JSP 中使用 JavaBean 组件等技术。本章作为 Web 开发的基础章节，为后续的 Web 应用程序的开发奠定基础。通过本章的学习，读者应该掌握以下基本内容：

- 如何创建一个 Web 应用程序
- 如何创建 HTML/XHTML 页面
- 如何创建 JSP 页面
- 如何使用各种 HTML 标准组件和 HTML 表单组件
- 如何在 JSP 中使用 JavaBean 组件，进行业务逻辑的分离

Download at http://www.pin5i.com/

第 12 章

高级 Web 应用程序开发

上一章介绍了如何使用 Applet 小程序以及 JSP 开发 Web 应用程序。本章将介绍如何开发 Servlet、使用标记等内容，以及如何进行网络监视、项目的打包部署、项目配置等。

在学习本章时，假定读者已经具有了编写 Servlet 等技术的能力，这里主要着重介绍如何在 NetBeans 集成开发环境中使用这些技术进行开发。对于一些初学者，建议首先学习必要的知识。另外，为了便于初学者学习，对于讲到的一些技术，书中都作了必要的解释，这样可以保证读者能够顺利地完成本章的学习。

本章主要内容

- 如何创建一个 Servlet
- 如何使用过滤器
- 标准标记库的开发
- 如何开发 Servlet

本章亮点

- 过滤器的使用方法
- 自定义标记库的开发
- 如何使用网络监听器

12.1 开发 Servlet

Servlet 是用 Java 编写的 Server 端程序，它使用了 Java Servlet 应用程序设计接口（API）及相关类和方法。Java Servlet 可以动态地扩展 Server 的能力，具有方便高效、功能强大、可移植性好、易于维护、性能稳定等优点。

12.1.1 在 NetBeans 中创建 Servlet

在 NetBeans 中可以快速地创建 Servlet。在创建 Web 标准项目后，选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”列表中选择“Web”节点，在右边的“文件类型”列表中选择 Servlet 节点，单击“下一步”按钮打开“新建 Servlet”对话框，如图 12-1 所示。

在“类名”文本框中输入 Servlet 类的名字，单击“下一步”按钮，打开“配置 Servlet 部署”对话框，如图 12-2 所示。选中“将消息添加到部署描述符（web.xml）”复选框，这样创建的 Servlet 将自动添加到 web.xml 文件中，否则就必须手工进行添加。

“Servlet 名称”文本框用于输入 Servlet 的名字，可以和类名不同。“URL 模式”文本框用于输入通过 URL 访问 Servlet 的路径，例如输入“/NewServlet”，那么就可以通过“http://your-server/your-web-project/NewServlet”访问该 Servlet。其中，your-server 为服务器路径；your-web-project 为 Web 项目的路径。

Download at http://www.pin5i.com/

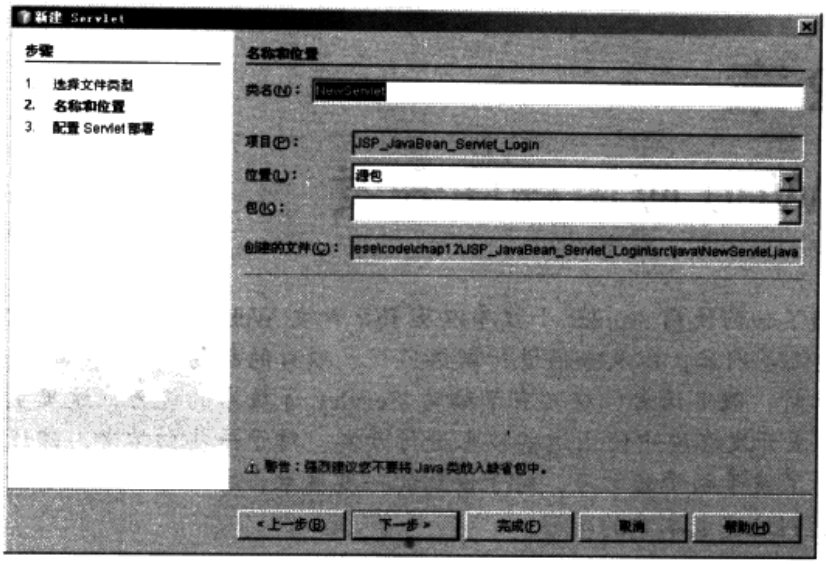


图 12-1 “新建 Servlet”对话框

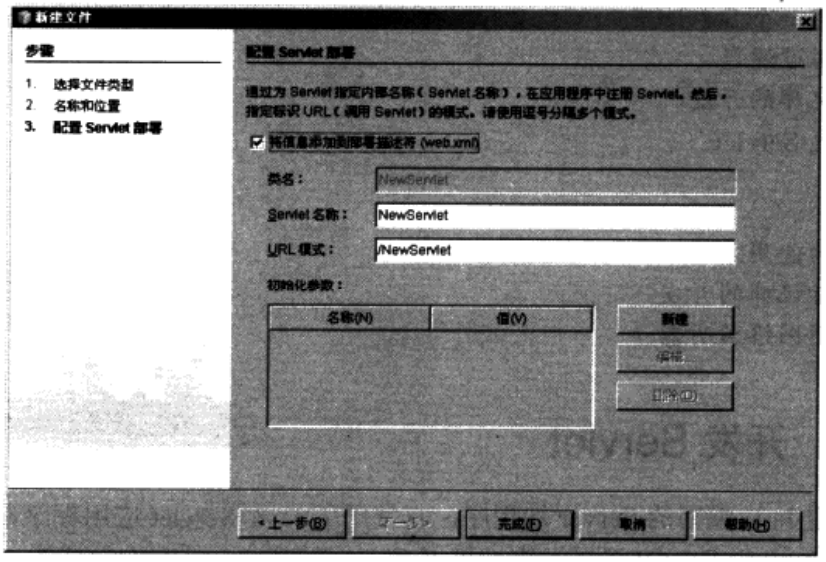


图 12-2 “配置 Servlet 部署”对话框

如果访问 Servlet 时需要指定参数，可以单击“新建”按钮进行添加。完成输入后，单击“完成”按钮，即可创建一个 Servlet。创建的 Servlet 代码如下所示：

```
/*
 * NewServlet.java
 *
 */

import java.io.*;
import java.net.*;

import javax.servlet.*;
import javax.servlet.http.*;

/**
 *
 * @author Liu Bin
 * @version
 */
```

Download at http://www.pin5i.com/


```
public class NewServlet extends HttpServlet {

    /** Processes requests for both HTTP <code>GET</code> and
     * <code>POST</code> methods.
     * @param request servlet request
     * @param response servlet response
     */
    protected void processRequest(HttpServletRequest request,
            HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        /* TODO output your page here
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet NewServlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet NewServlet at " +
                    request.getContextPath () + "</h1>");
        out.println("</body>");
        out.println("</html>");
         */
        out.close();
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods.
    // Click on the + sign on the left to edit the code.">
    /** Handles the HTTP <code>GET</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doGet(HttpServletRequest request,
            HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }

    /** Handles the HTTP <code>POST</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doPost(HttpServletRequest request,
            HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }

    /** Returns a short description of the servlet.
     */
    public String getServletInfo() {
        return "Short description";
    }
    // </editor-fold>
}
```

其中，doGet 方法用于处理 HTTP 的 GET 方法，doPost 方法用于处理 HTTP 的 POST 方法。如果有必要可以分开处理 GET 和 POST 方法。这里，在两个方法中都调用了 processRequest 方法进行集中处理。在 processRequest 方法中，生成了一个简单的 HTML 页面返回到客户端。

一般情况下，可以直接在 processRequest 方法中进行数据库访问，后者指定要显示的内容。实际上 Servlet 起了一个控制器的作用，集中处理客户端的请求。

Download at http://www.pin5i.com/



12.1.2 Servlet 使用实例——登录模块

上一节介绍了在 NetBeans 中创建 Servlet 的方法，本节将通过一个具体的登录模块的实例，来演示如何在 NetBeans 中进行 Servlet 开发。

在前面的章节中已经介绍了通过 JSP 和 JSP + JavaBean 实现的登录模块，本节将在 JSP + JavaBean 登录模块的基础上，通过 Servlet 实现登录模块。

❶ 创建一个名为 JSP_JavaBean_Servlet_Login 的 Web 应用程序，并设置为主项目。

❷ 从第 11.6.5 节创建的 JSP_JavaBean_Login 项目中，把 web 目录下的 index.jsp、error.jsp 和 welcome.jsp 文件，以及 images 子目录复制到当前项目的 web 目录下。再把 src\java 目录中的 login 子目录也复制到当前目录的相应位置。

❸ 按照上一节的方法创建一个包名为 login、类名为 LoginServlet 的 Servlet。在 processRequest 方法中实现如下的代码：

```
protected void processRequest(HttpServletRequest request,
            HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        //获取用户的输入
        String username = request.getParameter("username");
        if (username == null) {
            username = "";
        }
        String password = request.getParameter("password");
        if (password == null) {
            password = "";
        }

        //以下代码利用 LoginBean 进行登录判断
        boolean loginflag = false;
        LoginBean bean = new LoginBean();
        bean.setUsername(username);
        bean.setPassword(password);
        loginflag = bean.checklogin();

        //根据登录情况打开不同的页面（视图）
        if (loginflag) {
            request.getSession().setAttribute("username", username);
            response.sendRedirect("welcome.jsp");
        } else {
            response.sendRedirect("error.jsp");
        }
    }
```

❹ 为了能够让 Servlet 处理用户的请求，这里要把 index.jsp 文件中的下面两行代码：

```
<h1>请登录系统</h1>
<form action="logincheck.jsp" method="post">
```

替换为：

```
<h1>请登录系统</h1>
<form action="LoginServlet" method="post">
```

至此，完成了实现基于 Servlet 的登录模块开发。在这个实例中，用 Servlet 代替了 logincheck.jsp 文件的功能。在 Servlet 中通过 JavaBean 访问数据库，并根据访问结果实现不同的页面。这样进一步把界面和控制逻辑进行了分离，从而实现了一个基本的 MVC 框架。在下

Download at http://www.pin5i.com/

一章中，我们将会介绍使用 Struts 技术实现的登录模块，并进一步讲解 MVC 技术，以及 Struts 框架。

12.2 使用过滤器

在 Servlet 2.3 版中，一个重要的功能就是支持能够为 Servlet 和 JSP 页面定义过滤器。NetBeans 提供了对 Servlet 2.3 应用程序编程接口的支持，可以在集成开发环境中快速创建过滤器。

12.2.1 NetBeans 中创建过滤器的方法

在创建 Web 标准项目后，选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”列表中选择“Web”节点，在右边的“文件类型”列表中选择“过滤器”节点，单击“下一步”按钮打开“新建 过滤器”对话框，如图 12-3 所示。

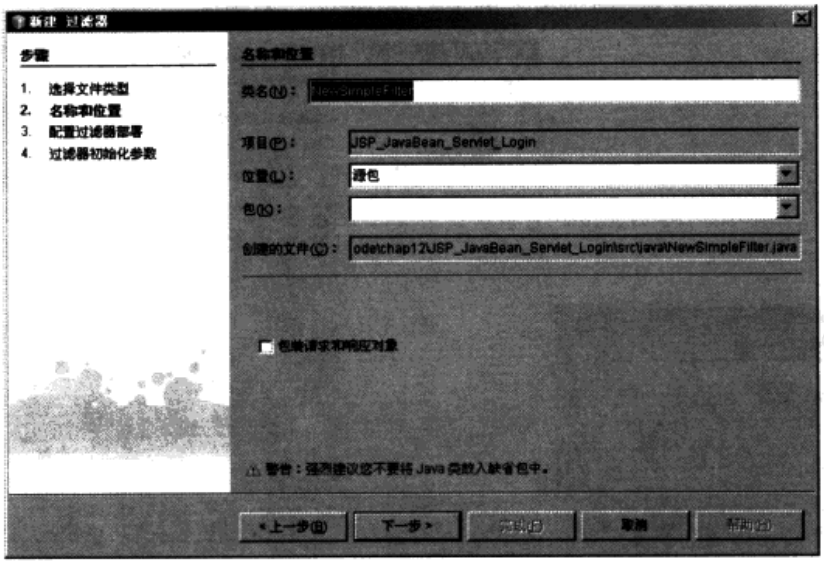


图 12-3 “新建 过滤器”对话框

在“类名”文本框中输入过滤器的名字。如果要对请求和响应对象进行包装，则需要选中“包装请求和响应对象”复选框。例如，可以使用以下方法扩展请求和响应的功能：

- 允许在将请求发送到过滤器链的其余部分之前，在请求上设置参数；
- 跟踪在响应上设置的 Cookie。

单击“下一步”按钮，打开“配置过滤器部署”对话框，如图 12-4 所示。选中“将消息添加到部署描述符（web.xml）”复选框，这样创建的过滤器将自动添加到 web.xml 文件中，否则就必须手工进行添加。“过滤器名称”文本框用于输入过滤器的名字，可以和类名不同。

单击“新建”按钮，可以打开“过滤器映射”对话框，如图 12-5 所示。选中“URL”单选按钮，然后在后面的文本框中输入要被过滤的地址。例如输入“/*”则表示项目目录下的所有文件的访问都要被过滤；如果输入“/data/*”，则要过滤 data 子目录下的所有文件访问。如果要过滤 Servlet，则要选中 Servlet 单选按钮，然后在后面的下拉列表框中选择要过滤的 Servlet。

单击“下一步”按钮，打开“过滤器初始化参数”对话框，如图 12-6 所示。单击“新建”按钮，可以为过滤器添加初始化的一些参数。

Download at http://www.pin5i.com/

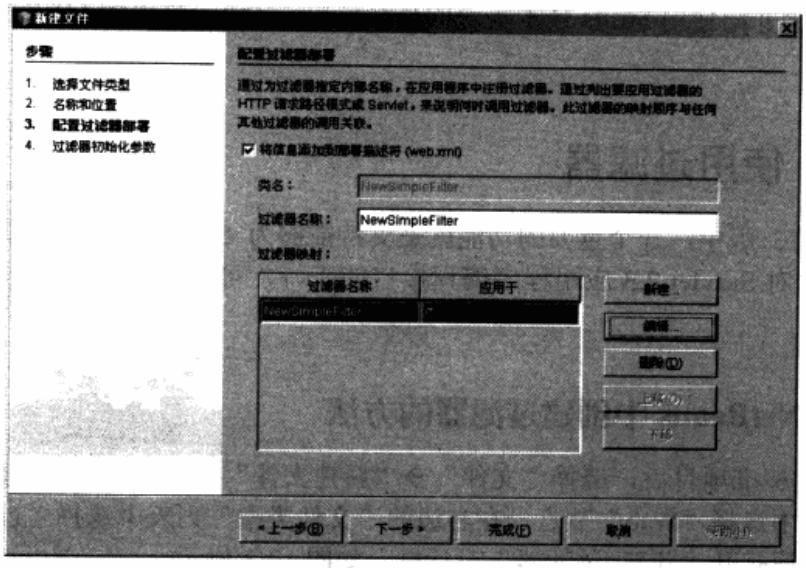


图 12-4 “配置过滤器部署”对话框

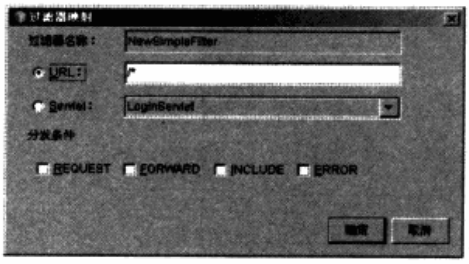


图 12-5 “过滤器映射”对话框

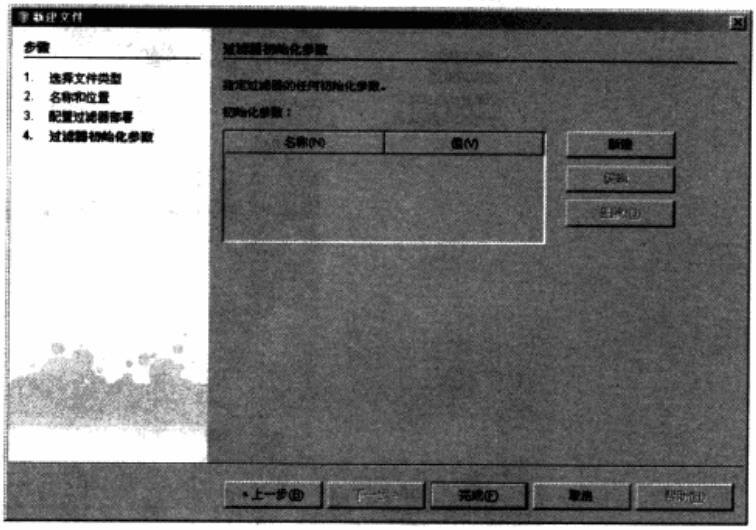


图 12-6 “过滤器初始化参数”对话框

单击“完成”按钮，创建过滤器，创建的过滤器代码如下所示：

```
/*
 * NewSimpleFilter.java
 *
 * Created on 2007年5月3日, 上午11:26
 */

import java.io.*;
import java.net.*;
import java.util.*;
import java.text.*;
import javax.servlet.*;
import javax.servlet.http.*;

import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
```

Download at http://www.pin5i.com/


```
import javax.servlet.ServletResponse;

/**
 *
 * @author  Liu Bin
 * @version
 */

public class NewSimpleFilter implements Filter {

    // The filter configuration object we are associated with.  If
    // this value is null, this filter instance is not currently
    // configured.
    private FilterConfig filterConfig = null;

    public NewSimpleFilter() {
    }

    private void doBeforeProcessing(ServletRequest request,
           ServletResponse response)
    throws IOException, ServletException {
        if (debug) log("NewSimpleFilter:DoBeforeProcessing");
        //
        // Write code here to process the request and/or response before
        // the rest of the filter chain is invoked.
        //

        //
        // For example, a logging filter might log items on the request object,
        // such as the parameters.
        //
        /*
         for (Enumeration en = request.getParameterNames();
                           en.hasMoreElements(); ) {
             String name = (String)en.nextElement();
             String values[] = request.getParameterValues(name);
             int n = values.length;
             StringBuffer buf = new StringBuffer();
             buf.append(name);
             buf.append("=");
             for(int i=0; i < n; i++) {
                 buf.append(values[i]);
                 if (i < n-1)
                     buf.append(",");
             }
             log(buf.toString());
         }
         */

    }

    private void doAfterProcessing(ServletRequest request,
                       ServletResponse response)
    throws IOException, ServletException {
        if (debug) log("NewSimpleFilter:DoAfterProcessing");
        //
        // Write code here to process the request and/or response after
        // the rest of the filter chain is invoked.
        //

        //
        // For example, a logging filter might log the attributes on the
        // request object after the request has been processed.
        //
        /*
        for (Enumeration en = request.getAttributeNames();
                          en.hasMoreElements(); ) {
            String name = (String)en.nextElement();
```

Download at http://www.pin5i.com/


```
            Object value = request.getAttribute(name);
            log("attribute: " + name + "=" + value.toString());

        }
         */
        //

        //
        // For example, a filter might append something to the response.
        //
        /*
        PrintWriter respOut = new PrintWriter(response.getWriter());
        respOut.println("<P><B>This has been appended by an intrusive filter.</B>");
         */
    }

    /**
     *
     * @param request The servlet request we are processing
     * @param result The servlet response we are creating
     * @param chain The filter chain we are processing
     *
     * @exception IOException if an input/output error occurs
     * @exception ServletException if a servlet error occurs
     */
    public void doFilter(ServletRequest request, ServletResponse response,
            FilterChain chain)
            throws IOException, ServletException {

        if (debug) log("NewSimpleFilter:doFilter()");

        doBeforeProcessing(request, response);

        Throwable problem = null;

        try {
            chain.doFilter(request, response);
        } catch(Throwable t) {
            //
            // If an exception is thrown somewhere down the filter chain,
            // we still want to execute our after processing, and then
            // rethrow the problem after that.
            //
            problem = t;
            t.printStackTrace();
        }

        doAfterProcessing(request, response);

        //
        // If there was a problem, we want to rethrow it if it is
        // a known type, otherwise log it.
        //
        if (problem != null) {
            if (problem instanceof ServletException)
                    throw (ServletException)problem;
            if (problem instanceof IOException) throw (IOException)problem;
            sendProcessingError(problem, response);
        }
    }


    /**
     * Return the filter configuration object for this filter.
     */
    public FilterConfig getFilterConfig() {
        return (this.filterConfig);
    }
```

Download at http://www.pin5i.com/


```
    /**
     * Set the filter configuration object for this filter.
     *
     * @param filterConfig The filter configuration object
     */
    public void setFilterConfig(FilterConfig filterConfig) {

        this.filterConfig = filterConfig;
    }

    /**
     * Destroy method for this filter
     *
     */
    public void destroy() {
    }


    /**
     * Init method for this filter
     *
     */
    public void init(FilterConfig filterConfig) {

        this.filterConfig = filterConfig;
        if (filterConfig != null) {
            if (debug) {
                log("NewSimpleFilter:Initializing filter");
            }
        }
    }

    /**
     * Return a String representation of this object.
     */
    public String toString() {

        if (filterConfig == null) return ("NewSimpleFilter()");
        StringBuffer sb = new StringBuffer("NewSimpleFilter(");
        sb.append(filterConfig);
        sb.append(")");
        return (sb.toString());

    }


    private void sendProcessingError(Throwable t, ServletResponse response) {

        String stackTrace = getStackTrace(t);

        if(stackTrace != null && !stackTrace.equals("")) {

            try {

                response.setContentType("text/html");
                PrintStream ps = new PrintStream(response.getOutputStream());
                PrintWriter pw = new PrintWriter(ps);
                pw.print("<html>\n<head>\n<title>Error</title>\n</head>\n<body>\n");
//NOI18N

                // PENDING! Localize this for next official release
                pw.print("<h1>The resource did not process correctly</h1>\n
<pre>\n");
                pw.print(stackTrace);
                pw.print("</pre></body>\n</html>"); //NOI18N
```

Download at http://www.pin5i.com/


```
                pw.close();
                ps.close();
                response.getOutputStream().close();;
            }

            catch(Exception ex){ }
        } else {
            try {
                PrintStream ps = new PrintStream(response.getOutputStream());
                t.printStackTrace(ps);
                ps.close();
                response.getOutputStream().close();;
            } catch(Exception ex){ }
        }
    }

    public static String getStackTrace(Throwable t) {

        String stackTrace = null;

        try {
            StringWriter sw = new StringWriter();
            PrintWriter pw = new PrintWriter(sw);
            t.printStackTrace(pw);
            pw.close();
            sw.close();
            stackTrace = sw.getBuffer().toString();
        } catch(Exception ex) {}
        return stackTrace;
    }

    public void log(String msg) {
        filterConfig.getServletContext().log(msg);
    }

    private static final boolean debug = true;
}
```

其中，在 doBeforeProcessing()方法中，可以添加过滤前的处理代码；在 doAfterProcessing()方法中，可以添加过滤后的处理代码；doFilter()方法是过滤的主要方法，过滤代码应该添加到这个方法中；init()方法是过滤器的初始化代码。

sendProcessingError()方法用于发送处理过程中的错误；getStackTrace()方法用于获得跟踪栈中的信息；Log()方法用于输出过滤记录；变量 debug 用于控制是否输出过滤记录，当值为 true 时（默认值）输出记录。



12.2.2 过滤器使用实例——为登录模块提供安全保护

在前面几个登录模块的实例中，都只介绍了如何进行登录校验，并没有增加任何的安全措施。事实上，如果用户知道要访问网页的地址，在不登录的情况下，直接访问这些页面地址都是可以“绕过”登录检验部分的。

本节将在第 12.1.2 节的实例基础上，利用过滤器为登录模块增加一个“防火墙”，把没有登录的非法访问进行屏蔽。

❶ 在“项目”窗口中，右键单击第 12.1.2 节创建的 JSP_JavaBean_Servlet_Login 项目节点，选择弹出菜单中的“复制项目”命令，打开“复制项目”对话框，如图 12-7 所示。

❷ 在“项目位置”文本框中选择新的项目路径，在“项目名称”文本框中输入“FilterDemo”，单击“复制”命令。

Download at http://www.pin5i.com/

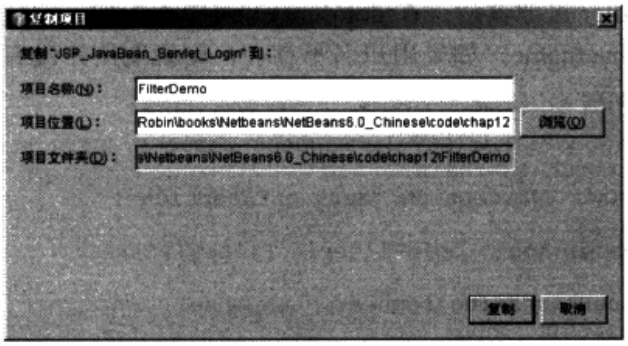


图 12-7 “复制项目”对话框

❸ 在新项目的 web 子目录下创建一个 data 子目录，后面创建的过滤器将对这个目录中所有文件的访问进行过滤，没有登录的访问将被屏蔽。在 data 子目录中，创建一个名为 ProtectedPage.jsp 的页面，用来模拟被保护的页面。其内容如下所示：

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>

    <h1>必须登录后才能查看</h1>

    你好，你已经登录了，谢谢！
    </body>
</html>
```

❹ 在 login 包中，按照上一小节的方法创建一个过滤器，过滤器的名字为 LoginFilter。在如图 12-8 所示的对话框中，修改过滤器映射的路径为“/data/*”。这样，只过滤对 data 子目录下任何资源的访问。

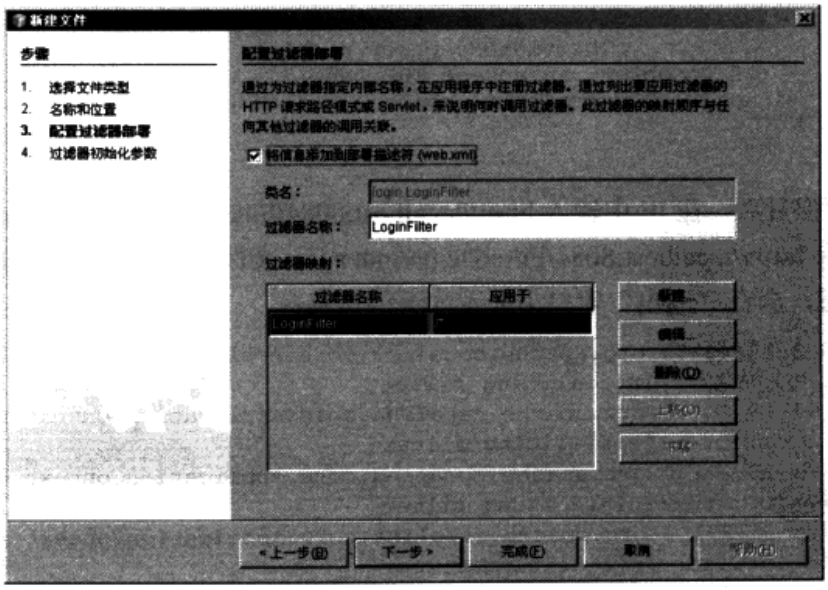


图 12-8 修改过滤器映射的路径

Download at http://www.pin5i.com/

❺ 在创建 LoginFilter 过滤器后，在 doFilter()方法中添加如下的代码，进行过滤。首先从 Session 中读取用户名 username，如果用户名不存在，则说明用户没有登录，那么就把页面重定向到登录页面 index.jsp。

```
public void doFilter(ServletRequest request, ServletResponse response,
        FilterChain chain)
        throws IOException, ServletException {

    if (debug) log("LoginFilter:doFilter()");

    doBeforeProcessing(request, response);

    Throwable problem = null;

    HttpServletRequest httprequest = (HttpServletRequest)request;
    HttpServletResponse httpresponse = (HttpServletResponse)response;
    HttpSession session = httprequest.getSession();
    String username = (String)session.getAttribute("username");

    //登录界面不需要过滤
    if (username == null || username == "") {
        httpresponse.sendRedirect("../index.jsp");
        if (debug) {
            log("被拦截一个未认证的请求");
        }
    } else {
        try {
            chain.doFilter(request, response);
        } catch(Throwable t) {
            problem = t;
            t.printStackTrace();
        }
    }

    doAfterProcessing(request, response);

    //
    // If there was a problem, we want to rethrow it if it is
    // a known type, otherwise log it.
    //
    if (problem != null) {
        if (problem instanceof ServletException) throw
                (ServletException)problem;
        if (problem instanceof IOException) throw (IOException)problem;
        sendProcessingError(problem, response);
    }
}
```

❻ 在部署项目后，在 IE 中输入 http://localhost:8084/FilterDemo/data/ProtectedPage.jsp，回车后马上会显示 http://localhost:8084/FilterDemo/index.jsp 文件。这表示非法的的页面访问已经被屏蔽了。此时，会在“输出”窗口中显示如下的信息：

```
2007-5-3 15:19:25 org.apache.catalina.core.ApplicationContext log
信息: LoginFilter:Initializing filter
2007-5-3 15:20:45 org.apache.catalina.core.ApplicationContext log
信息: LoginFilter:Initializing filter
2007-5-3 15:22:30 org.apache.catalina.core.ApplicationContext log
信息: LoginFilter:Initializing filter
2007-5-3 15:29:16 org.apache.catalina.core.ApplicationContext log
信息: LoginFilter:Initializing filter
2007-5-3 15:37:13 org.apache.catalina.core.ApplicationContext log
信息: LoginFilter:Initializing filter
2007-5-3 15:37:23 org.apache.catalina.core.ApplicationContext log
```

Download at http://www.pin5i.com/


```
信息: LoginFilter:Initializing filter
2007-5-3 15:38:30 org.apache.catalina.core.ApplicationContext log
信息: LoginFilter:doFilter()
2007-5-3 15:38:30 org.apache.catalina.core.ApplicationContext log
信息: LoginFilter:DoBeforeProcessing
2007-5-3 15:38:30 org.apache.catalina.core.ApplicationContext log
信息: 被拦截一个未认证的请求
2007-5-3 15:38:30 org.apache.catalina.core.ApplicationContext log
信息: LoginFilter:DoAfterProcessing
```

如果登录后，再访问 ProtectedPage.jsp，则可以正常访问该页面。这表示过滤器按照要求过滤掉了未授权的页面请求。

12.3 创建侦听程序

在初始化或销毁 Servlet 上下文和会话时，以及在上下文或会话中添加或删除属性时，侦听程序都会通过生命周期事件获知这些操作。

在创建标准的 Web 应用程序后，选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”列表中选择“Web”节点，在右边的“文件类型”列表中选择“Web 应用程序侦听程序”节点，单击“下一步”按钮打开“新建 Web 应用程序侦听程序”对话框，如图 12-9 所示。

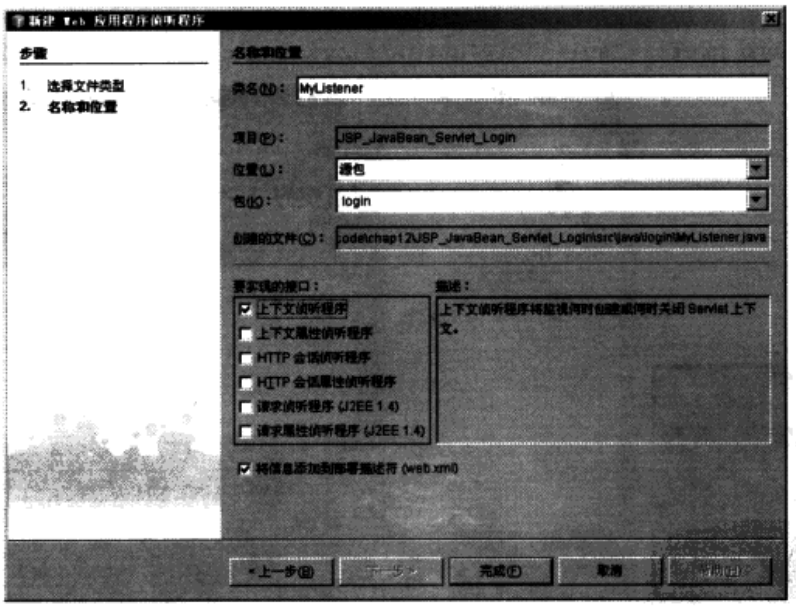


图 12-9 “新建 Web 应用程序侦听程序”对话框

在“类名”文本框中输入监听器的名字。在“要实现的接口”中，可以选中不同的复选框，以监听不同的内容。单击“完成”按钮，即可创建监听器。

12.4 使用标准标记库

JSP 的一个优势就是可以使用标记库，加快程序的开发。JSP 标准标记库 JSTL 全称为 Java ServerPages Standard Tag Library，是一个不断完善的开源 JSP 标签库。本节将主要介绍 NetBeans

Download at http://www.pin5i.com/

中使用 JSTL 库的方法，下一节将介绍如何自己开发标记库。

12.4.1 JTSL 概述以及 NetBeans 对标准标记库的支持

JSTL 是一个已经被标准化的标记库集合，其开发 API 规范可以在 www.jcp.org 下载。JSTL 主要包括如表 12-1 所示的四个基本标记库。

表 12-1 JSTL 标记库集合

库 名 字	说 明	库 路 径
Core 核心库	支持输出、设置、迭代、条件等	http://java.sun.com/jstl/core
XML 处理库	主持 XML 的处理	http://java.sun.com/jstl/xml
I18N 页面的国际化库	支出页面的国际化操作	http://java.sun.com/jstl/fmt
SQL 数据库处理库	支持对数据库进行 SQL 操作	http://java.sun.com/jstl/sql sql

创建 JSP 页面后，可以在 NetBeans 组件面板中看到如图 12-10 所示的 JSTL 标记库。其中，JSTL Choose、JSTL If、JSTL For Each 包含在核心库中；数据库分类下的几个标记包含在数据库处理库中。

如果要在项目中使用 JSTL 库，那么需要在项目的库路径中添加 JSTL 库的路径，如图 12-11 所示。同时要在使用 JSTL 标记的页面开始的地方加上库路径的引用。下面第一行代码是对核心库的引用；第二行代码是对数据库处理库的引用：

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
```

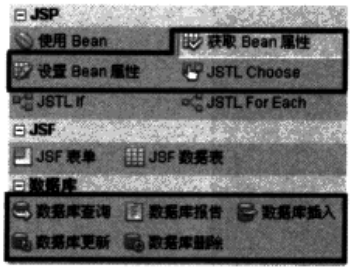


图 12-10 NetBeans 中的 JSTL

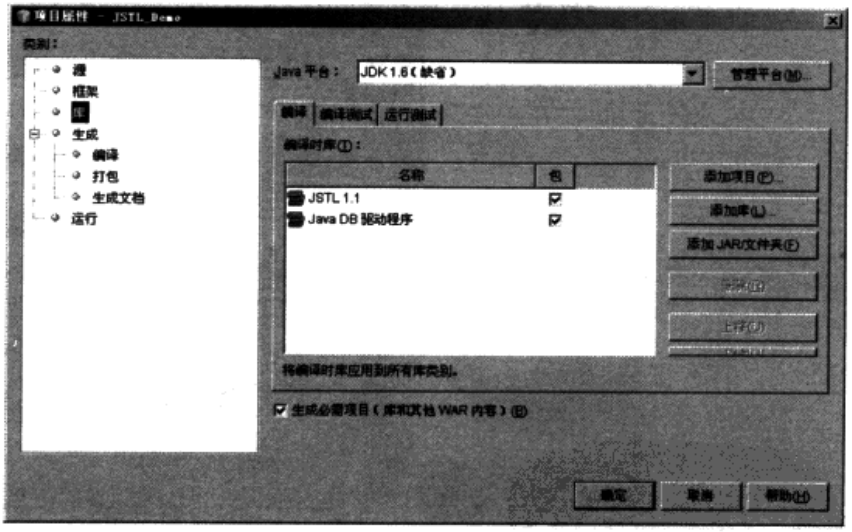


图 12-11 在项目中添加 JSTL 库路径

12.4.2 使用核心库标记

核心库标记包括<c:out>、<c:set>、<c:choose>、<c:if>、<c:each>等标记，分别用于输出信息、设置值、多项选择、选择和迭代，下面分别进行介绍。

1. JSTL If 标记

NetBeans 工具组件面板中的 JSTL If 图标用于插入<c:if></c:if>标记，进行条件判断。在创建了 JSP 页面后，从 NetBeans 工具组件面板中，拖曳 JSTL If 图标到 JSP 页面中。释放鼠标后

Download at http://www.pin5i.com/

会显示如图 12-12 所示的“插入 JSTL If”对话框。

在“条件”文本框中输入判断的条件，例如${param.name != null}，用于判断页面请求 URL 中名字为 name 的参数是否存在。“变量名称”文本框用于设置测试条件结果值的变量名。“范围”文本框用于设置作用范围，可以有如下的几个选项。

- page：当前页面范围内有效。
- request：当前请求范围内有效。
- session：当前会话范围内有效。
- application：当前应用程序生命周期范围内有效。

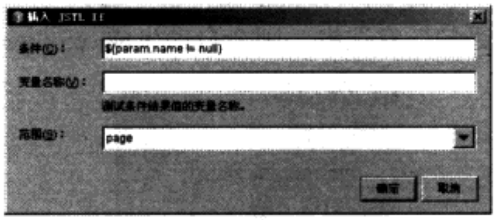


图 12-12 “插入 JSTL If”对话框

图 12-12 所示的对话框将会自动创建如下的代码：

```
            <c:if test="${param.name !=
null}">
            </c:if>
```

下面代码是一个简单的<c:if></c:if>标记的使用演示，其作用是显示 URL 链接中名为 name 的参数值。

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>

        <h1>JSTL If</h1>

        <c:if test="${param.name != null}">
            <!-- 欢迎用户 ${param.name} -->
            Hello ${param.name}!
        </c:if>

    </body>
</html>
```

当通过 http://localhost:8084/JSTL_Demo/JSTL_if.jsp 链接访问该页面时，显示界面如图 12-13 所示。当通过 http://localhost:8084/JSTL_Demo/JSTL_if.jsp?name=Robin 链接访问该页面时，显示界面如图 12-14 所示。

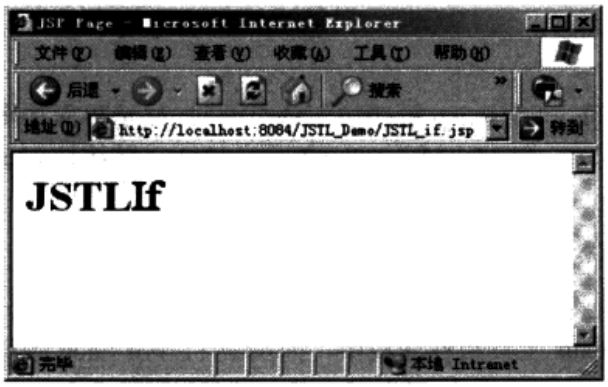


图 12-13 不带参数时的访问界面

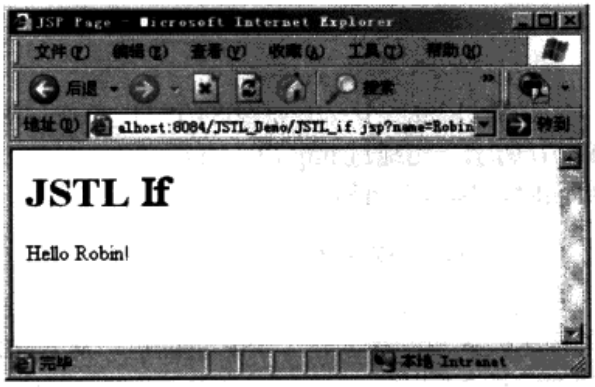


图 12-14 带参数时的访问界面

Download at http://www.pin5i.com/

2. JSTL Choose 标记

NetBeans 工具组件面板中的 JSTL Choose 图标用于插入<c:choose></c: choose >标记，进行多个条件判断。在创建了 JSP 页面后，从 NetBeans 工具组件面板中，拖曳 JSTL Choose 图标到 JSP 页面中。释放鼠标后会显示如图 12-15 所示的“插入 JSTL Choose”对话框。

其中，“When 条件数”微调文本框用于设置要处理的条件数。如果选中“生成 Otherwise 语句”复选框，则会自动生成<c:otherwise></c:otherwise>语句。图 12-15 所示的对话框将会自动创建如下的代码：

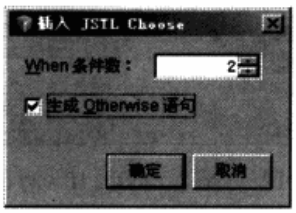


图 12-15 “插入 JSTL Choose”对话框

```
<c:choose>
        <c:when test="">
        </c:when>
        <c:when test="">
        </c:when>
        <c:otherwise>
        </c:otherwise>
    </c:choose>
```

下面代码是一个简单的<c:choose></c: choose >标记的使用演示，其作用根据 URL 链接中名为 name 的参数值判断用户的身份。当用户为“Robin”时就显示系统管理员身份；当用户为“Apple”时就显示超级用户身份；其他用户显示为一般用户。

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>

        <h1>JSTL Choose</h1>
        <c:choose>
            <c:when test="${param.name == 'Robin'}">
                <font color="blue">${param.name}是管理员身份
            </c:when>
            <c:when test="${param.name == 'Apple'}">
                <font color="red">${param.name}是超级用户身份
            </c:when>
            <c:otherwise>
                <font color="green">${param.name} 一般用户身份
            </c:otherwise>
        </c:choose>

    </body>
</html>
```

当通过 http://localhost:8084/JSTL_Demo/JSTL_choose.jsp 链接访问该页面时，显示界面如图 12-16 所示。当通过 http://localhost:8084/JSTL_Demo/JSTL_ choose.jsp?name=Robin 链接访问该页面时，显示界面如图 12-17 所示。

3. JSTL For Each 标记

NetBeans 工具组件面板中的 JSTL For Each 图标用于插入<c: forEach></c: forEach>标记，进行迭代处理。在创建了 JSP 页面后，从 NetBeans 工具组件面板中，拖曳 JSTL For Each 图标到 JSP 页面中。释放鼠标后会显示如图 12-18 所示的“插入 JSTL For Each”对话框。

Download at http://www.pin5i.com/

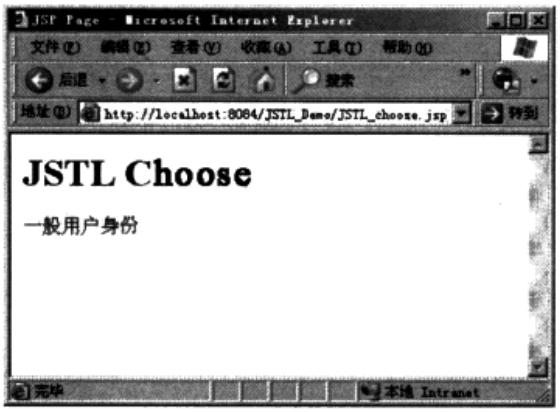


图 12-16 不带参数时的访问界面

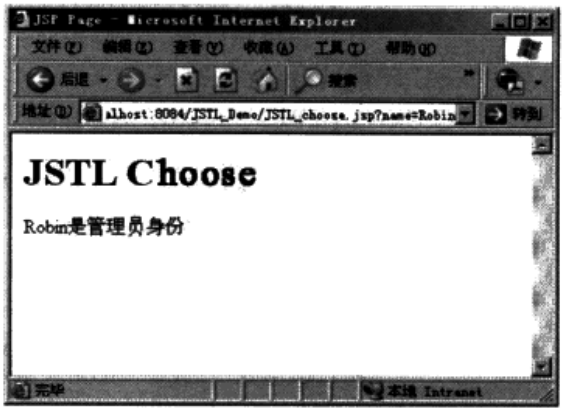


图 12-17 带参数时的访问界面

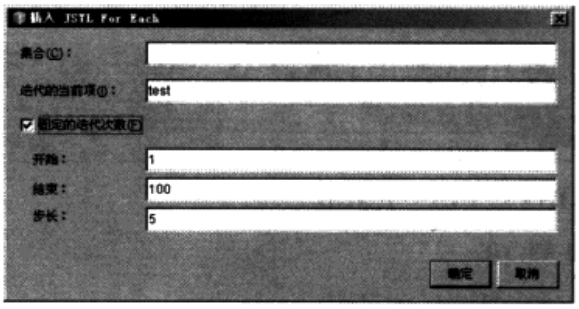


图 12-18 “插入 JSTL For Each”对话框

其中，“集合”文本框用于输入要进行循环的集合或者集合名字（如果已经在其他地方定义了集合）。“迭代的当前项”文本框用于输入迭代变量的名字。forEach 标记支持两种类型的迭代：在集合中迭代或者指定一个固定的范围。如果选中“固定的迭代次数”复选框，那么可以不需要输入迭代集合。在“开始”文本框中输入迭代的起始值；在“结束”文本框中输入迭代的结束值；如果迭代步长不为 1，还需要在“步骤”文本框中输入迭代变量的变化步长。图 12-18 所示的对话框将会自动创建如下的代码：

```
<c:forEach var="test" items="" begin="1" end="100" step="5">
</c:forEach>
```

下面代码是一个简单的<c: forEach ></c: forEach >标记的使用演示。代码分为三个主要部分：第一个部分演示固定迭代从 1 到 10，步长为 1；第二个部分演示固定迭代从 1 到 100，步长为 5；第三个部分演示在集合中迭代。集合的定义通过<c:set />标记实现，var 部分定义了集合的名字，value 部分给出了集合的值。在 forEach 标记中通过${colors}引用集合。在迭代的过程中，通过<c:out /> 标记输出当前迭代变量 token 的值${token}。程序的运行界面如图 12-19 所示。

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>JSTL For Each</h1>

        <h2>1 到 10</h2>
        <c:forEach var="i" begin="1" end="10">
```

Download at http://www.pin5i.com/


```
            ${i} &#149;
        </c:forEach>

        <p></p>

        <h2>1 到 100 步长 5</h2>
        <c:forEach var="i" begin="1" end="100" step="5">
            ${i} &#149;
        </c:forEach>

        <p></p>

        <h2>集合</h2>
        <c:set var="colors" value="red, blue, yellow, black, white" />
        <c:forEach var="token" items="${colors}">
            <c:out value="${token}"/><br>
        </c:forEach>
    </body>
</html>
```



图 12-19 forEach 标记演示

12.4.3 使用数据库处理标记

核心库标记包括< <sql:setDataSource>、<sql:update>、<sql:query>、<sql:transaction>等标记，分别用于设置数据库数据源、进行数据更新或者数据库操作、进行数据查询、创建数据库事务，下面分别进行介绍。

1. 创建表和删除

在进行任何的数据库操作前，应该首先设置数据库数据源。数据源的设置通过<sql:setDataSource>标记实现，下面是一个访问 Java DB 数据库 MyTest 的设置参数。其中，var 定义了数据源的名字，以便于在其他地方引用该数据源；driver 定义了数据库的驱动程序，如果为其他数据库，请替换为相应数据库的驱动程序；url 为数据库访问的 URL 地址；user 是登录数据库的用户名；password 是登录数据库的用户密码。

```
<sql:setDataSource
    var="mytest"
    driver="org.apache.derby.jdbc.ClientDriver"
    url="jdbc:derby://localhost:1527/MyTest"
```

Download at http://www.pin5i.com/


```
    user="admin"
    password="111"
/>
```

要创建数据表，可以从 NetBeans 工具组件面板中，拖曳“数据库更新”图标到 JSP 页面中。释放鼠标后会显示如图 12-20 所示的“插入 数据库更新”对话框。

其中，“变量名称”文本框用于数据更新变量的名字。“范围”下拉列表框用于设置作用范围。“数据源”文本框用于设置要使用的数据源，这里的${mytest}就是前面定义的数据源。在“Update 语句”文本框中可以输入任何的更新语句，例如更新表、创建表、更新记录等。下面的代码用于创建一个名字为 mytable 的数据表：

```
create table mytable (
nameid int primary key,
name varchar(80)
)
```

图 12-20 所示的对话框将会自动创建如下的代码：

```
<sql:update var="newTable" dataSource="${mytest}">
    create table mytable (
    nameid int primary key,
    name varchar(80)
    )
</sql:update>
```

要删除数据表，只要使用 drop table 语句即可，这里不再赘述。

2. 数据的插入和更新

要进行数据更新，可以从 NetBeans 工具组件面板中，拖曳“数据库插入”图标到 JSP 页面中。释放鼠标后会显示如图 12-21 所示的“插入数据库插入”对话框。

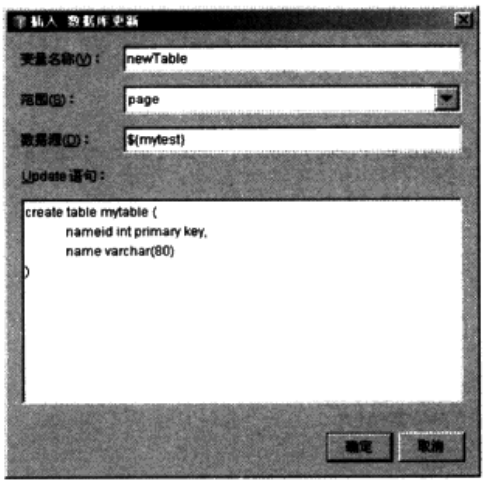


图 12-20 “插入 数据库更新”对话框

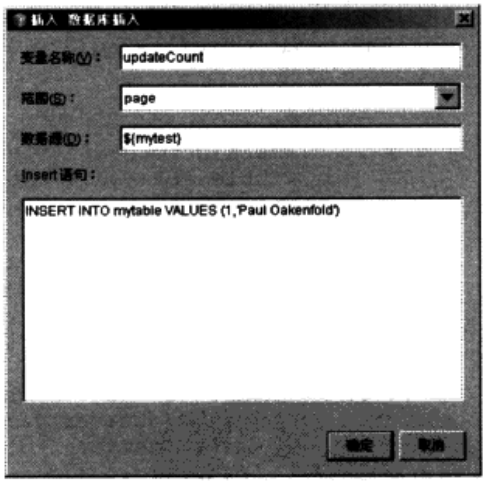


图 12-21 “插入数据库插入”对话框

在“Insert 语句”文本框中输入进行插入和更新的 SQL 语句，例如输入如下的语句即可在 mytable 表中插入一条语句。

```
INSERT INTO mytable VALUES (1,'Paul Oakenfold')
```

图 12-21 所示的对话框将会自动创建如下的代码：

```
<sql:update var="updateCount" dataSource="${mytest}">
    INSERT INTO mytable VALUES (1,'Paul Oakenfold')
</sql:update>
```

Download at http://www.pin5i.com/

数据的更新可以使用 NetBeans 工具组件面板中的“数据库更新”标记，SQL 语句为 Update 语句，这里不再赘述。



图 12-22 “插入 数据库查询”对话框

3. 数据查询

要进行数据查询，可以从 NetBeans 工具组件面板中，拖曳“数据库查询”图标到 JSP 页面中。释放鼠标后会显示如图 12-22 所示的“插入 数据库查询”对话框。

在“查询语句”文本框中输入进行查询的 SQL 语句，例如输入如下的语句，即可显示 mytable 表中的所有内容。

```
SELECT * FROM mytable
```

图 12-22 所示的对话框将会自动创建如下的代码：

```
<sql:query var="mytable"
dataSource="${mytest}">
    SELECT * FROM mytable
</sql:query>
```

4. 数据库标记使用实例

下面的代码在 Java DB 数据库 MyTest 中，首先创建了一个 mytable 数据表，然后插入 3 行数据并通过 SQL 语句进行查询，查询的结果以三种不同的方式进行迭代和显示。最后，把创建的 mytable 表删除。

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>

        <h1>数据库标记演示</h1>

        <sql:setDataSource
            var="mytest"
            driver="org.apache.derby.jdbc.ClientDriver"
            url="jdbc:derby://localhost:1527/MyTest"
            user="admin"
            password="111"
        />
        <sql:transaction dataSource="${mytest}">
            <!-- 创建数据表-->
            <sql:update var="newTable">
                create table mytable (
                nameid int primary key,
                name varchar(80)
                )
            </sql:update>

            <!-- 插入数据-->
            <sql:update var="updateCount">
                INSERT INTO mytable VALUES (1,'Paul Oakenfold')
            </sql:update>
```

Download at http://www.pin5i.com/


```
    <sql:update var="updateCount">
        INSERT INTO mytable VALUES (2,'Timo Maas')
    </sql:update>
    <sql:update var="updateCount">
        INSERT INTO mytable VALUES (3,'Paul van Dyk')
    </sql:update>

    <!-- 数据查询-->
    <sql:query var="mytable">
        SELECT * FROM mytable
    </sql:query>
</sql:transaction>

<p></p>

<!-- 显示查询结果 1-->
<h2>通过行索引迭代</h2>
<table border="1">
    <c:forEach var="row" items="${mytable.rowsByIndex}">
        <tr>
            <c:forEach var="column" items="${row}">
                <td><c:out value="${column}"/></td>
            </c:forEach>
        </tr>
    </c:forEach>
</table>

<p></p>

<!-- 显示查询结果 2-->
<h2>通过行集合迭代</h2>
<table border="1">
    <c:forEach var="row" items="${mytable.rows}">
        <tr>
            <td>Name: <c:out value="${row.NAMEID}"/></td>
            <td>Value: <c:out value="${row.NAME}"/></td>
        </tr>
    </c:forEach>
</table>

<p></p>

<!-- 显示查询结果同时显示字段名字-->
<h2>Putting it all together</h2>
<table border="1">
    <tr>
    <%-- 使用列名字作为表头 --%>
    <c:forEach var="columnName" items="${mytable.columnNames}">
        <th><c:out value="${columnName}"/></th>
    </c:forEach>

    <%-- 显示表中的内容 --%>
    <c:forEach var="row" items="${mytable.rowsByIndex}">
        <tr>
        <c:forEach var="column" items="${row}">
            <td><c:out value="${column}"/></td>
        </c:forEach>
    </c:forEach>
</table>

<%-- 删除表 --%>
<sql:update var="newTable" dataSource="${mytest}">
    drop table mytable
</sql:update>
```

Download at http://www.pin5i.com/


```
    </body>
</html>
```

以上代码的运行界面如图 12-23 所示。



图 12-23 数据库标记的演示界面

12.4.4 JSTL 开发实例——使用 JSTL 实现登录模块

前面两小节介绍了如何使用 JSTL 标记库，下面通过使用这个标记库来实现登录模块。创建一个 Web 应用程序，名字为 Login_Tag，并设置为主项目。把第 11 章中创建的 JSP_Login 项目的 web 目录下的四个 JSP 文件以及 images 子目录复制到当前项目的 web 目录下。

把 logincheck.jsp 文件的内容修改为如下内容，通过 SQL 标记库来访问数据库并判断是否登录成功。

```
<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<c:set var="password" value="${param.password}"/>
<c:set var="username" value="${param.username}"/>
<c:set var="u" value="admin"/>
<c:set var="p" value="111"/>
<c:set var="loginsuccess" value="false"/>

<sql:setDataSource
    var="mytest"
    driver="org.apache.derby.jdbc.ClientDriver"
    url="jdbc:derby://localhost:1527/MyTest"
    user="${u}"
    password="${p}"
/>

<sql:query var="login" dataSource="${mytest}">
    SELECT "Password" FROM USERS
    WHERE "UserName"=<c:out value="'${username}'" escapeXml="false"/>
</sql:query>

<c:forEach var="row" items="${login.rows}">
```

Download at http://www.pin5i.com/


```
        <c:if test="${row.PASSWORD == password}">
            <c:set var="loginsuccess" value="true"/>
        </c:if>
    </c:forEach>

    <c:choose>
        <c:when test="${loginsuccess=='true'}">
            <c:set value="${username}" var="username" scope="session"/>
            <c:redirect url="welcome.jsp"/>
        </c:when>
        <c:otherwise>
            <c:redirect url="error.jsp"/>
        </c:otherwise>
    </c:choose>
```

以上代码中，语句<c:out value="'${username}'" escapeXml="false"/>用于获得加单引号的用户名，其中 escapeXml="false"是必须的，否则单引号会变为“#039;”符号。

在项目的库路径中添加了 JSTL 以及 Java DB 库后，就可以运行项目了。其运行结果和以前所示的一致。

12.5 自定义标记开发

除了使用标准的标记库外，开发人员也可以根据自己的需要开发自定义的标记使用。在 NetBeans 中可以通过编写标记文件，或者通过编写称为标记处理程序的 Java 类来实现自定义标记。

12.5.1 通过标记文件实现标记开发

在 NetBeans 中可以创建一个空的标记文件，通过这个标记文件可以创建定制的 JSP 标记。使用标记文件实现标记的好处就是，不需要编写 Java 类。创建的标记文件应该放置在 Web 应用程序的/WEB-INF/tags 目录下。下面通过一个实例演示通过标记文件实现标记开发的过程。

❶ 创建一个名字为 UserTagFile 的 Web 应用程序，并设置为主项目。

❷ 打开“新建文件”对话框，选择“类别”列表中的“Web”节点，再选择“文件类型”列表的“标记库描述符”节点，创建一个标记库描述符 TLD（Tag Library Descriptor）文件。标记库描述符文件是一个包含库相关信息的 XML 文档，该文档还包含库中的标记和标记文件的相关信息。创建的标记要被添加到标记库描述符文件中，在 JSP 文件中通过标记库描述符来引用标记。

❸ 单击“下一步”按钮，打开“新建 标记库描述符”对话框，如图 12-24 所示。

❹ 在“TLD 名称”文本框中输入描述符文件的名字，这里输入 mytagdesp。在“URI”文本框中将显示该文件的路径，这个路径将在后面使用到。“前缀”文本框中是标记要使用的前缀，这里使用默认的值即可。

❺ 单击“完成”按钮，创建标记库描述符文件，文件内容如下所示。其中，<short-name>用于指定自定义标记描述的名字；<uri>用于指定自定义标记描述文件的路径。

```
    <?xml version="1.0" encoding="UTF-8"?>
    <taglib version="2.0" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-jsptaglibrary_2_0.xsd">
      <tlib-version>1.0</tlib-version>
      <short-name>mytagdesp</short-name>
      <uri>/WEB-INF/tlds/mytagdesp</uri>
      <!-- A validator verifies that the tags are used correctly at JSP
           translation time. Validator entries look like this:
        <validator>
```

Download at http://www.pin5i.com/


```
            <validator-class>com.mycompany.TagLibValidator</validator-class>
            <init-param>
               <param-name>parameter</param-name>
               <param-value>value</param-value>
    </init-param>
       </validator>
     -->
    <!-- A tag library can register Servlet Context event listeners in
         case it needs to react to such events. Listener entries look
         like this:
      <listener>
          <listener-class>com.mycompany.TagLibListener</listener-class>
      </listener>
    -->
</taglib>
```

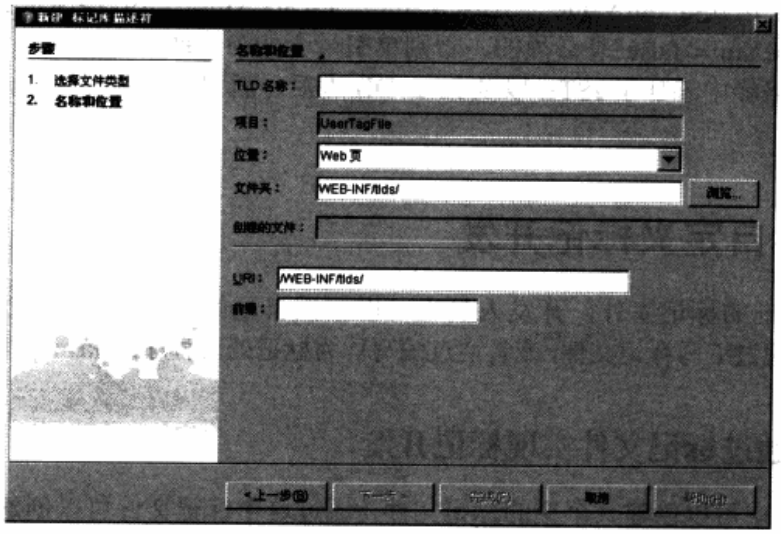


图 12-24 “新建 标记库描述符”对话框

❻ 打开“新建文件”对话框，选择“类别”列表中的“Web”节点，再选择“文件类型”列表的“标记文件”节点，创建一个空的新标记文件。单击“下一步”按钮，打开“新建 标记文件”对话框，如图 12-25 所示。

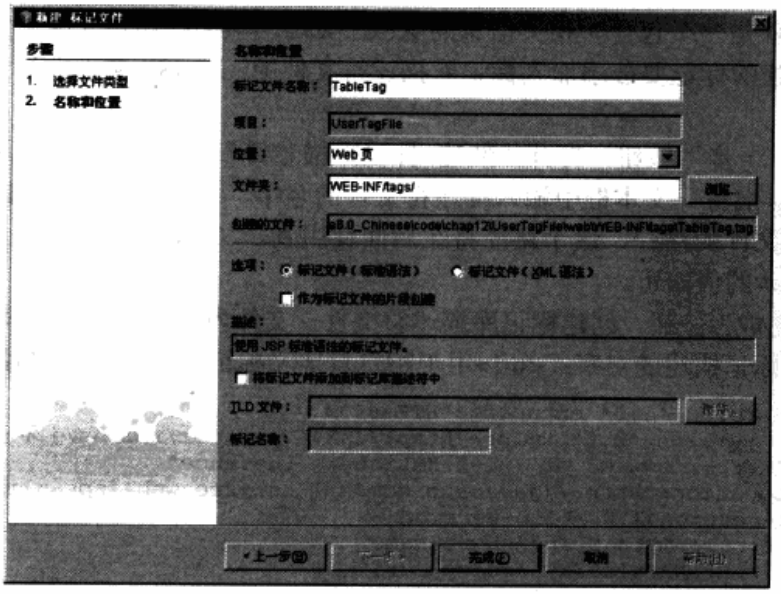


图 12-25 “新建 标记文件”对话框

Download at http://www.pin5i.com/

❼ 在“标记文件名称”文本框中输入标记文件的名字“TableTag”。选中“将标记文件添加到标记库描述符中”复选框。单击“TLD 文件”文本框旁边的“浏览”按钮，选择刚才创建的标记库描述文件。

❽ 单击“完成”按钮，创建一个空标记文件，修改内容如下所示。该标记文件用于显示一个表格，表格内容为图书信息，由参数 title、author 和 price_info 指定。

```
<%@tag description="My Table" pageEncoding="UTF-8"%>

<%@attribute name="title" required="true"%>
<%@attribute name="author"%>
<%@attribute name="price_info" %>

<table border="1">
    <tr>
        <td align="center"><h2>${title}</h2></td>
    </tr>
    <tr>
        <td><em>作者：</em><strong>${author}</strong></td>
    </tr>
    <tr>
        <td><em>价格：</em><strong>${price_info}</strong></td>
    </tr>
</table>
```

❾ 返回标记库描述文件，删除其中的注释，代码变为如下内容。其中，<tag-file>用于说明一个标记；<name>用于说明标记的名字；<path>用于说明标记的路径。

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib version="2.0" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-jsptaglibrary_2_0.xsd">
  <tlib-version>1.0</tlib-version>
  <short-name>mytagdesp</short-name>
  <uri>/WEB-INF/tlds/mytagdesp</uri>

  <tag-file>
    <name>TableTag</name>
    <path>/WEB-INF/tags/TableTag.tag</path>
  </tag-file>
</taglib>
```

❿ 修改 index.jsp 文件为如下的内容，使用创建的标记文件来显示两个表格。表格内容为图书《精通 NetBeans》和《精通 Java ME 开发》的详细信息。

```
<%@ page contentType="text/html; charset=GBK" %>
<%@ taglib prefix="table" uri="WEB-INF/tlds/mytagdesp.tld" %>
<html>
    <head>
        <title>
            helloworld
        </title>
    </head>
    <body bgcolor="#ffffff">
        <h1>
            使用创建的标记文件
        </h1>

        <table:TableTag author="Robin" title="精通 NetBeans" />
        <table:TableTag author="Apple" title="精通 Java ME 开发"
        price_info="54.6"/>
    </body>
</html>
```

Download at http://www.pin5i.com/

⓫ 运行应用程序，显示界面如图 12-26 所示。



图 12-26 使用标记文件演示界面

12.5.2 通过标记处理程序实现标记开发

在通过标记处理程序开发标记时，需要创建一个标记处理文件，然后在该文件中定义标记的处理方法。自定义标记的名称及属性会自动添加到标记库描述库文件中。

使用自定义标记时，通过标记库描述符来引用标记，标记的处理通过标记处理程序来实现。下面创建一个用于登录的标记，然后通过这个标记来实现登录模块。

❶ 创建一个名为“Login_MyTag”的 Web 应用程序，并设置为主项目。把上一章中创建的 JSP_JavaBean_Login 项目的 web 目录下的四个 JSP 文件及 images 子目录复制到当前项目的 web 目录下。把 JSP_JavaBean_Login 项目的 src\java 目录下的 login 目录复制到当前项目的对应位置。

❷ 按照第 12.5.1 节中的方法，创建一个名为“Login.tld”的标记库描述符文件。

❸ 打开“新建文件”对话框，选择“类别”列表中的“Web”节点，再选择“文件类型”列表的“标记处理程序”节点，创建一个新的标记处理程序类。标记处理程序是服务器端的 Java 组件，用于创建 JSP 文件的定制标记。

❹ 单击“下一步”按钮，打开“新建 标记处理程序”对话框，如图 12-27 所示。

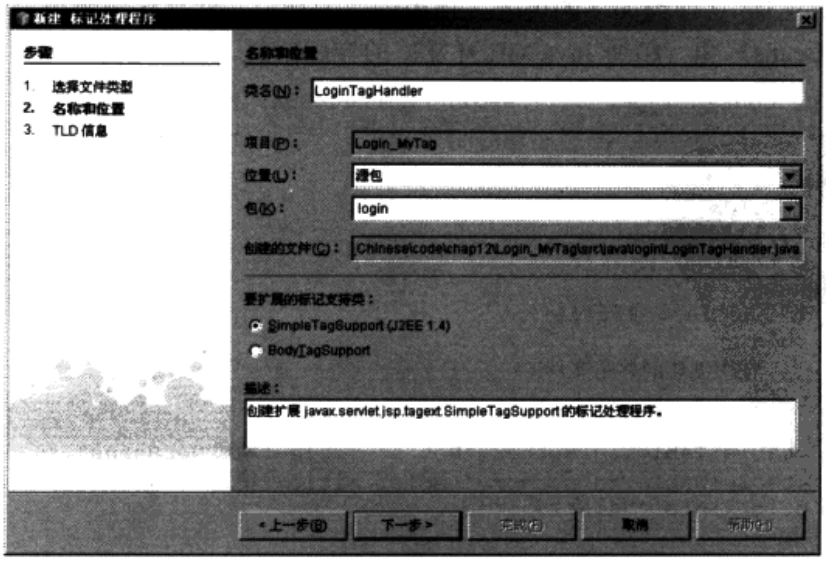


图 12-27 “新建 标记处理程序”对话框

Download at http://www.pin5i.com/

❺ 在“类名”文本框中输入标记处理程序类的类名“LoginTagHandler”，“包”选择“login”包。要扩展的标记支持类有两个选项：如果选中 SimpleTagSupport (J2EE 1.4)单选按钮，那么创建的标记处理程序类会继承于 javax.servlet.jsp.tagext.SimpleTagSupport；如果选中 BodyTagSupport 单选按钮，那么创建的标记处理程序类会继承于 javax.servlet.jsp.tagext.BodyTagSupport。这里选中 SimpleTagSupport (J2EE 1.4)单选按钮，单击“下一步”按钮，打开“TLD 信息”对话框，如图 12-28 所示。

❻ 选中“将相应标记添加到标记库描述符”复选框，单击“浏览”按钮，选择标记库描述符文件。单击“新建”按钮，打开“编辑 属性”对话框，如图 12-29 所示。在“属性名称”文本框中输入属性的名字，这里输入“username”。在“属性类型”下拉列表框中选择属性的类型，这里选择“java.lang.String”。选中“在请求时计算值”单选按钮。单击“确定”按钮，添加属性。

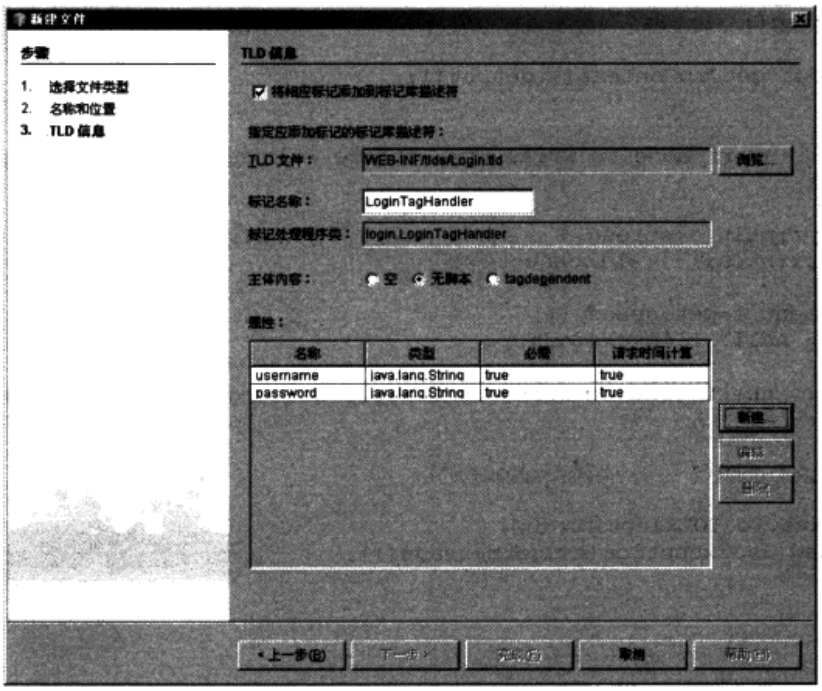


图 12-28 “TLD 信息”对话框

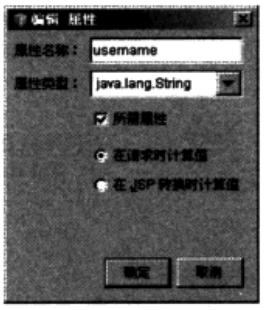


图 12-29 “编辑 属性”对话框

❼ 按照上一步的方法再创建一个类型为 java.lang.String、名字为 password 的属性。如图 12-28 中“属性”列表所示。单击“完成”按钮，创建标记处理程序。创建的标记处理程序的代码如下所示。

```
/*
 * LoginTagHandler.java
 *
 * Created on 2007年5月26日, 上午10:29
 */

package login;

import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.JspException;

/**
 *
 * @author  Liu Bin
 * @version
```

Download at http://www.pin5i.com/


```
 */
public class LoginTagHandler extends SimpleTagSupport {

    /**
     * Initialization of username property.
     */
    private java.lang.String username;

    /**
     * Initialization of password property.
     */
    private java.lang.String password;

    /**Called by the container to invoke this tag.
     * The implementation of this method is provided by the tag library developer,
     * and handles all tag processing, body iteration, etc.
     */
    public void doTag() throws JspException {

        JspWriter out=getJspContext().getOut();

        try {
            // TODO: insert code to write html before writing the body content.
            // e.g.:
            //
            // out.println("<strong>" + attribute_1 + "</strong>");
            // out.println("    <blockquote>");

            JspFragment f=getJspBody();
            if (f != null) f.invoke(out);

            // TODO: insert code to write html after writing the body content.
            // e.g.:
            //
            // out.println("    </blockquote>");

        } catch (java.io.IOException ex) {
            throw new JspException(ex.getMessage());
        }

    }

    /**
     * Setter for the username attribute.
     */
    public void setUsername(java.lang.String value) {
        this.username = value;
    }

    /**
     * Setter for the password attribute.
     */
    public void setPassword(java.lang.String value) {
        this.password = value;
    }
}
```

❽ 修改 doTag 方法，增加标记的处理代码如下所示。代码中通过 LoginBean 来进行登录判断，登录的结果保存在 success 属性中。

```
public void doTag() throws JspException {
    login.LoginBean login = new login.LoginBean();
    login.setUsername(this.username);
    login.setPassword(this.password);
    boolean success = login.checklogin();
    if (success) {
```

Download at http://www.pin5i.com/


```
            getJspContext().setAttribute("success", "true");
        } else {
            getJspContext().setAttribute("success", "false");
        }
    }
```

❾ 修改标记库描述符文件，删除不需要的注释，文件内容如下所示。可以看到，这里通过<attribute>定义了两个属性 username 和 password。至此，已经创建了一个自定义的标记了。下面将使用这个标记进行登录判断。

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib version="2.0" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-jsptaglibrary_2_0.xsd">
  <tlib-version>1.0</tlib-version>
  <short-name>login</short-name>
  <uri>/WEB-INF/tags/Login</uri>
  <tag>
    <name>LoginTagHandler</name>
    <tag-class>login.LoginTagHandler</tag-class>
    <body-content>scriptless</body-content>
    <attribute>
      <name>username</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
      <type>java.lang.String</type>
    </attribute>
    <attribute>
      <name>password</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
      <type>java.lang.String</type>
    </attribute>
  </tag>
</taglib>
```

❿ 修改 logincheck.jsp 文件，通过自定义的标记进行登录。登录的结果通过${pageScope.success}变量值进行判断。

```
<%@ taglib prefix="login" uri="WEB-INF/tlds/Login.tld" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<c:set var="password" value="${param.password}"/>
<c:set var="username" value="${param.username}"/>

<login:LoginTagHandler username = "${username}" password="${password}" />

<c:set var="loginsuccess" value="${pageScope.success}"/>
<c:choose>
    <c:when test="${loginsuccess=='true'}">
        <c:set value="${username}" var="username" scope="session"/>
        <c:redirect url="welcome.jsp"/>
    </c:when>
    <c:otherwise>
        <c:redirect url="error.jsp"/>
    </c:otherwise>
</c:choose>
```

至此，完成了通过自定义标记进行登录的模块开发，其运行结果和其他几种方法的结果完全一样，说明自定义标记工作正常。

Download at http://www.pin5i.com/

12.6 使用 HTTP 监视器

NetBeans 集成开发环境提供了一个 HTTP 监视器，用于帮助诊断 Web 服务器上来自 JSP 页和 Servlet 执行的数据流问题。在开发具有联网功能的 Java ME 移动应用程序时，还可以通过 HTTP 监视器，帮助进行程序的监视和分析。

12.6.1 打开 HTTP 监视器

默认情况下，在把应用程序部署到 Tomcat Web Server 时将自动启动 HTTP 监视器。在 NetBeans 集成开发环境中，选择“窗口”→“HTTP 监视器”命令，或者按下 Ctrl + Shift + 5 组合键，都可以手工打开 HTTP 监视器。

对于其他一些服务器，例如 Glass Fish V2，可以在“运行环境”窗口中展开“服务器”节点，右键单击“服务器”节点，选择弹出菜单中的“属性”命令，打开服务器管理器，如图 12-30 所示。

如果选中“启用 HTTP 监视器”复选框，那么在把应用程序部署到该服务器时会自动打开 HTTP 监视器。

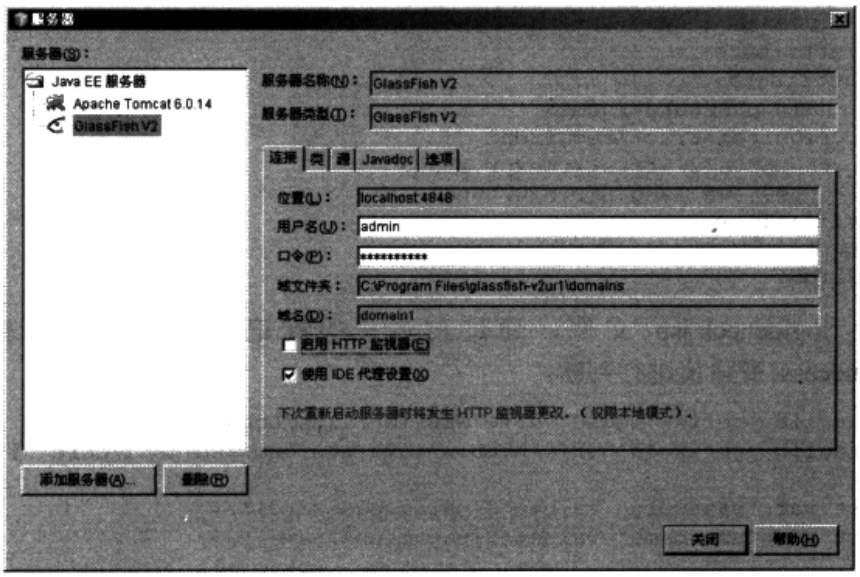


图 12-30 服务器管理器

12.6.2 分析会话数据

HTTP 监视器的显示界面如图 12-31 所示，它由两个部分组成。HTTP 监视器左边的部分为一个树型视图，列出了所有当前的会话记录，以及已经保存的会话记录。

“当前记录”中的条目仅在当前会话期间可用。当前 HTTP 请求记录不受重新启动服务器的影响。只有退出 NetBeans 集成开发环境或显式删除它们时，才会清除这些记录。“保存记录”中的条目在删除之前将始终保留。

每个记录都有一个图标，用于表示 HTTP 请求的方法。图标表示该记录使用的是 GET 方法；图标表示该记录使用的是 POST 方法。

在树形列表的上面是一个工具栏，工具栏上的按钮说明如表 12-2 所示。

Download at http://www.pin5i.com/

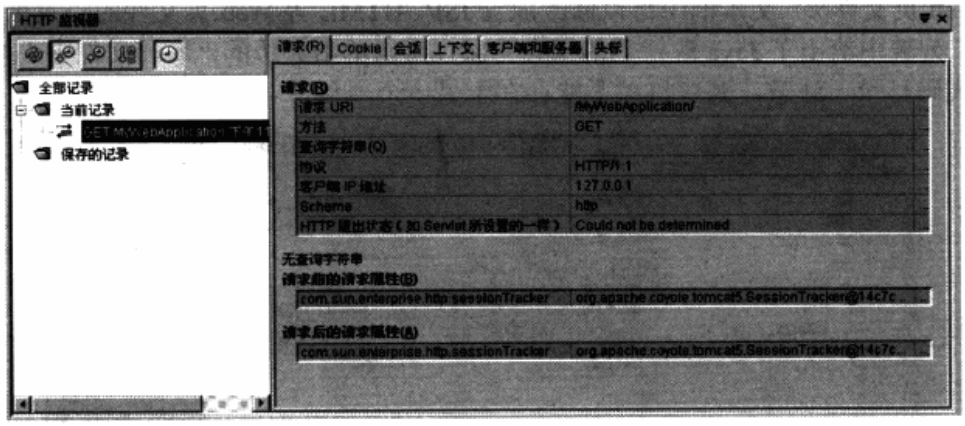


图 12-31 HTTP 监视器

表 12-2 HTTP 监视器工具栏上的按钮说明

图 标	描 述
	重新装入客户端上存储的所有 HTTP 请求记录
	按时间标记对 HTTP 请求记录进行降序排序
	按时间标记对 HTTP 请求记录进行升序排序
	按字母顺序对 HTTP 请求记录进行排序
	在 HTTP 请求记录列表中显示或隐藏时间标记

当在左侧的树型视图中选中某个记录时，与该 HTTP 请求记录相对应的会话数据将会显示在右侧的显示面板中。具体内容包括以下几个部分。

- 请求：显示了请求相关的信息，例如请求方法、请求 URI、使用的协议、客户端 IP 地址等内容。
- Cookie：传入和传出的 Cookie 信息。
- 会话：会话相关的属性，如会话 ID、会话创建时间、会话的上次访问时间等。
- 上下文：上下文以及 Servlet 上下文信息。
- 客户端和服务器：客户端和服务器端的相关信息。
- 头标：HTTP 的头部信息。

注意：树型视图中“当前记录”节点中的 HTTP 请求记录不会自动保存，右键单击要保存的 HTTP 请求记录（可以使用 Shift 或 Ctrl 键一次选择多个记录），选择弹出菜单中的“保存”命令，选定记录将被保存并移动到“保存的记录”子类别中。

12.7 Web 应用程序项目配置

Web 应用程序和标准项目类似，也可以在“项目属性”对话框中进行项目配置。在“项目”视图中，右键单击要配置的 Web 项目根节点，选择弹出菜单中的“属性”命令，可以打开“项目属性”对话框。本节将详细介绍项目配置的一些具体参数。

12.7.1 配置项目源和 Web 文件夹

在“项目属性”对话框左边的“类别”视图中选择“源”节点，显示界面如图 12-32 所示。

Download at http://www.pin5i.com/

在“Web 页文件夹”文本框中可以修改放置 JSP、HTML 等 Web 页文件的目录名字。在“源包文件夹”列表中给出了 Java 源代码的保存目录，可以单击旁边的“添加文件夹”按钮，添加其他的源代码目录。注意，源包目录不能存在于另一个项目中。

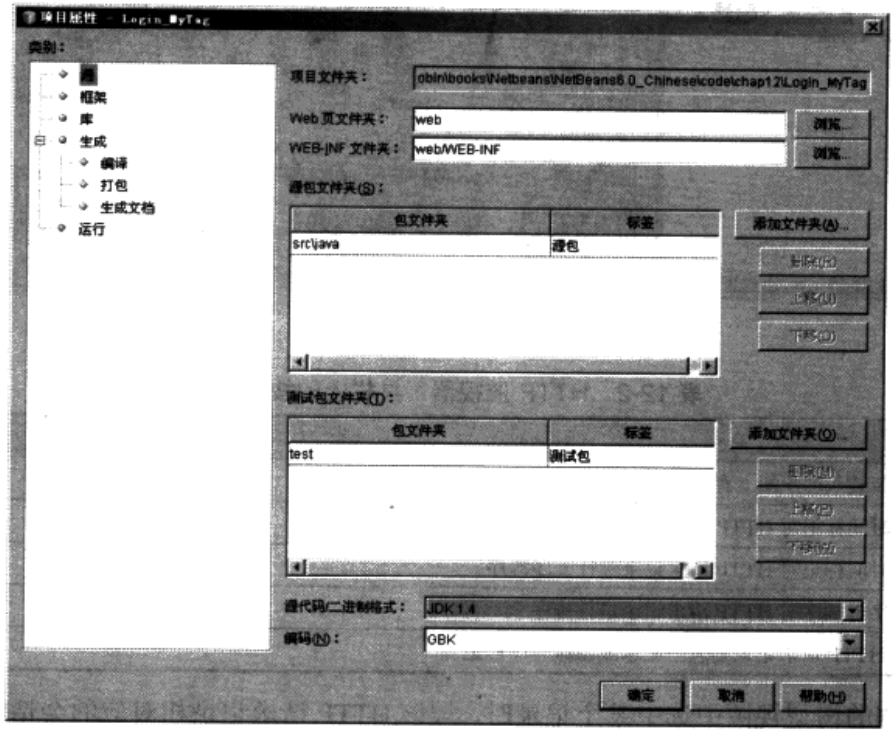


图 12-32　配置项目源和 Web 文件夹

在“测试包文件夹”列表中给出了测试代码的保存目录，可以单击旁边的“添加文件夹”按钮，添加其他的测试代码目录。

在“源代码/二进制格式”下拉列表框中可以选择源代码的级别，设置与生成的 Java 代码兼容的 Java 平台版本。可以选择 1.2、1.3、1.4、1.5 等几个级别，此设置对应于 javac -source 选项。指定源代码级别并不会更改 Java 平台。

“编码”下拉列表框用来设置页面的编码，默认为 UTF-8。

12.7.2　为项目添加新框架

在“项目属性”对话框左边的“类别”视图中选择“框架”节点，可以设置项目中使用的框架。单击“添加”按钮，可以打开如图 12-33 所示的“添加框架”对话框。在该对话框中列出了两个可用的框架：Vistual Web JavaServer Faces 和 JavaServer Faces。如果希望在已经创建的项目中增加新的框架，可以在这里选择一个框架，然后单击“确定”按钮进行添加。

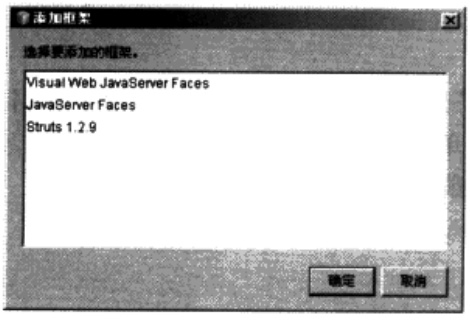


图 12-33　“添加框架”对话框

12.7.3　指定项目的类路径

在“项目属性”对话框左边的“类别”视图中选择“库”节点指定项目的类路径，如图 12-34 所示。

在“Java 平台”下拉列表框中可以指定编译和运行所有源文件时使用的目标 Java 平台。可用的 Java 平台是指那些已在 Java 平台管理器中注册的平台。

Download at http://www.pin5i.com/

选择“编译”标签，可以指定编译项目时使用的库。单击“添加项目”按钮，可以添加一个项目。单击“添加库”按钮，可以打开如图 12-35 所示的“添加库”对话框，用于选择一个库。单击“添加 JAR/文件夹”按钮，可以添加一个单独的 JAR 文件或者一个包含库文件的文件夹。

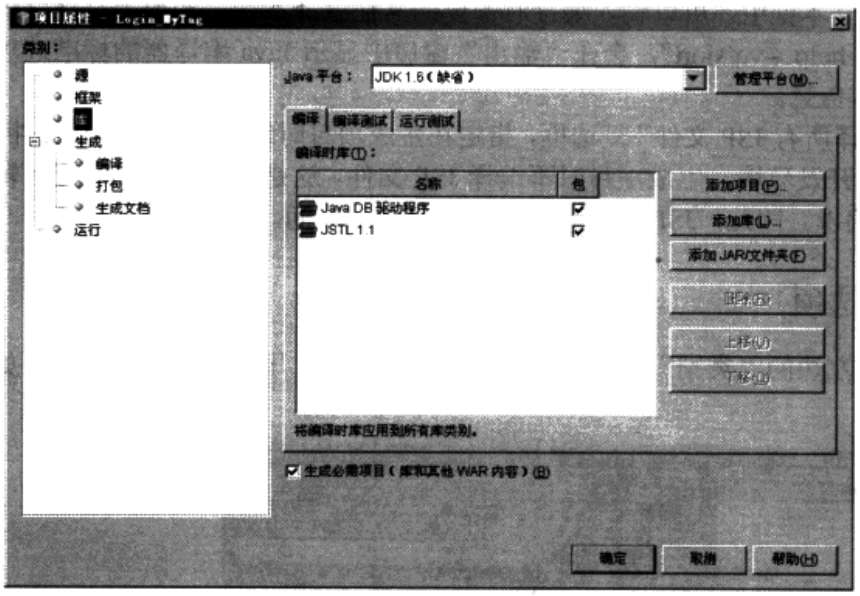


图 12-34 指定项目的类路径

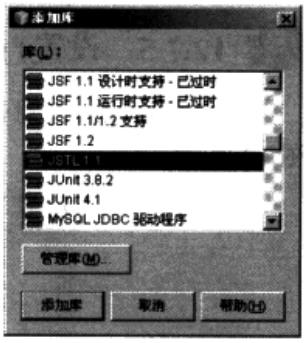


图 12-35 “添加库”对话框

单击“删除”按钮，可以删除一个选中的库。单击“上移”按钮，可以把选中的库的位置向上移动一个位置。

选中“生成必需项目（库和其他 WAR 内容）”复选框，在生成当前项目的同时还会生成当前项目所需的项目。必需项目是指类路径中的项目以及在“打包”页中定义的项目。如果不希望在生成当前项目时生成必需项目，请取消该复选框。



12.7.4 设置编译选项

在“项目属性”对话框左边的“类别”视图中选择“编译”节点，用来设置编译选项，如图 12-36 所示。

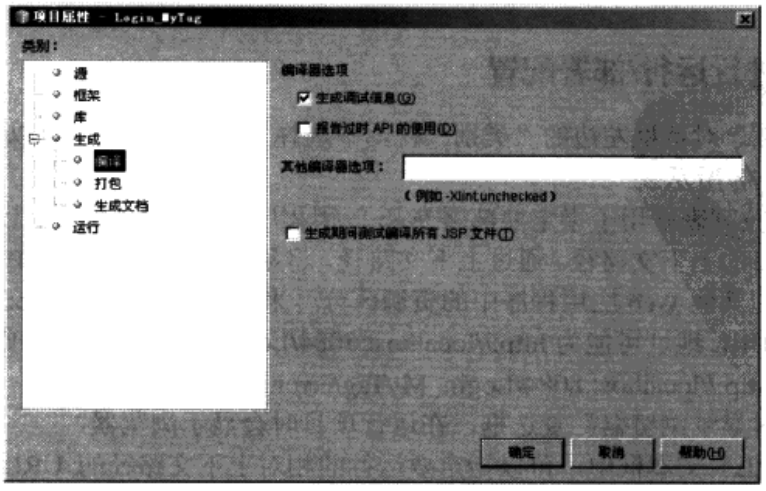


图 12-36 设置编译选项

Download at http://www.pin5i.com/

选中“生成调试信息”复选框，可以在编译项目时同时生成调试信息。如果要调试项目，则需要选中该复选框。在编译正式发布项目时，可以去掉该选项。

选中“报告过时 API 的使用”复选框，可以在编译的时候报告已经过时不用的 API，以增加项目的可移植性。

在“其他编译器选项”文本框中，可以输入编译时需要的其他编译参数。参数为空格分隔的列表形式，例如，键入“-help –version”，会在“输出”窗口中显示 Java 编译器的标准选项摘要以及 Java 编译器的版本信息。

选中“生成期间测试编译所有 JSP 文件”复选框，指定在生成项目时编译项目的 JSP 文件。默认情况下，NetBeans 集成开发环境在生成项目时不编译 JSP 文件。这是因为 JSP 文件必须转换为 Servlet 后，才能进行编译。因此，JSP 编译过程可能需要很长时间。

12.7.5 设置打包选项

在“项目属性”对话框左边的“类别”视图中选择“打包”节点，用来设置打包选项，如图 12-37 所示。

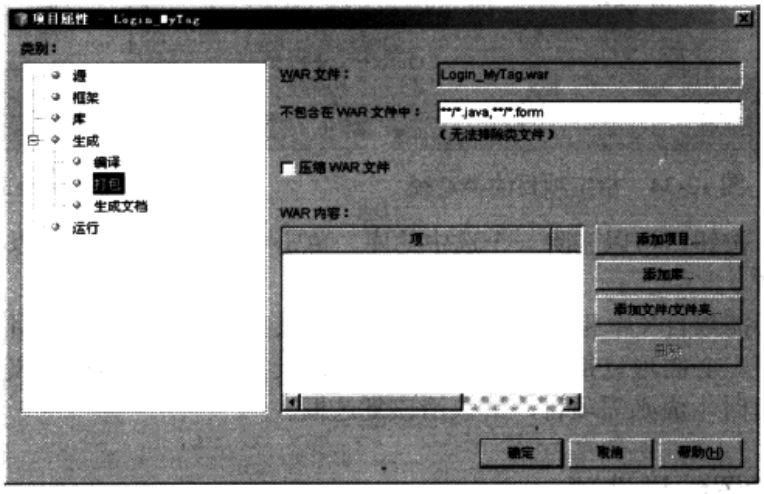


图 12-37 设置打包选项

在“不包含在 WAR 文件中”文本框中可以设置不希望打包到 WAR 文件中的文件类型。选中“压缩 WAR 文件”复选框，可以对生成的 WAR 文件进行压缩。

12.7.6 进行运行/部署配置

在“项目属性”对话框左边的“类别”视图中选择“运行”节点，可以配置用于部署项目的选项，如图 12-38 所示。

“服务器”下拉列表框用于指定要部署 Web 应用程序所在的服务器。“上下文路径”文本框用于指定运行项目的上下文路径。通过上下文路径，将某个 Web 应用程序中的资源与部署在同一服务器实例上的其他 Web 应用程序中的资源区分开来。例如，对于上下文“/Login_MyTag”，那么访问项目的 URL 地址可能为 http://localhost:8084/Login_MyTag。对于项目中的 error.jsp 文件，那么路径为 http://localhost:8084/Login_MyTag/error.jsp。

选中“运行时显示浏览器”复选框，在运行项目时会显示浏览器。

在“相对 URL”文本框中，可以指定要运行的相对上下文路径的 URL。例如，如果要将 /admin/login.jsp 文件作为 Web 应用程序的入口点，可以在这里输入/admin/login.jsp。

Download at http://www.pin5i.com/

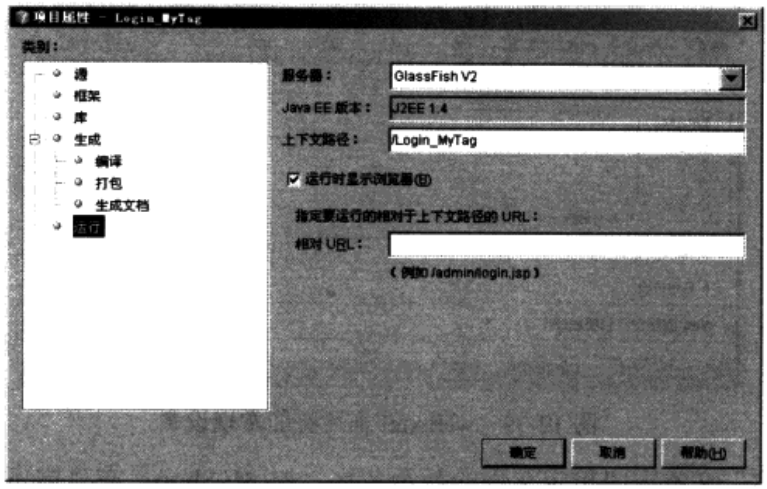


图 12-38 进行运行/部署配置

12.8 配置 context.xml 描述符

部署描述符是基于 XML 的文本文件，其元素对如何组装模块以及如何将模块部署到特定环境中进行了描述。这些元素还包含有关未直接包括在代码中的组件的行为信息。对于标准的 Web 应用程序，有以下三种类型的部署描述符。

- context.xml：特定于服务器的部署描述符，用于配置 Tomcat Web Server 的部署设置。
- web.xml：常规 Web 应用程序部署描述符，用于配置属于 Web 应用程序的组件的部署设置，如 Servlet 和 JSP 文件。
- sun-web.xml：特定于服务器的部署描述符，用于配置 Sun Java System Application Server 的部署设置。

本节介绍 context.xml 描述符配置方法，web.xml 和 sun-web.xml 的配置方法将在后两节中进行介绍。

在“项目”窗口中，展开 Web 应用程序项目的“配置文件”节点。双击 context.xml，将在源代码编辑器中打开 context.xml。下面是一个典型的 context.xml 内容，其中<Context path>标记定义了应用程序的上下文路径。该值和第 12.7.6 节中设置的上下文路径值一致。

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/Login_MyTag"/>
```

12.9 配置 web.xml 描述符

在“项目”窗口中，展开 Web 应用程序项目的“配置文件”节点。双击 web.xml，可以在 web.xml 可视编辑器中将其打开并进行编辑。打开 XML 标签，可以直接对 web.xml 文件进行编辑。

12.9.1 常规

在 web.xml 可视编辑器中选中“常规”标签，展开“常规”节点，可以定义 web.xml 文件的一些常规元素，如图 12-39 所示。

Download at http://www.pin5i.com/

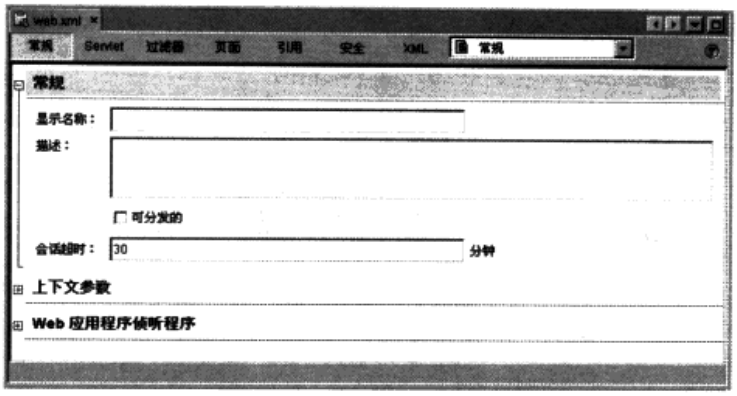


图 12-39 web.xml 描述符的常规设置

在“显示名称”文本框中指定文件的显示名称，如 MyShop。在“描述”文本框中可以指定应用程序的描述信息。

选中“可分发的”复选框，可以指定会话中的所有请求必须同时由一个 Java 虚拟机来处理。在“会话超时”文本框中，可以为当前 Web 应用程序中创建的所有会话指定默认会话超时间隔，单位为分钟。如果会话超时时间设置为-1，则会话将永不过期。

12.9.2 上下文参数

上下文参数是可在整个 Web 应用程序中使用的初始化参数，利用上下文参数，可以初始化一些有可能经常变动的内容，以增加页面的灵活性。

在 web.xml 可视编辑器中选中“常规”标签，展开“上下文参数”节点，来添加、删除和查看 Web 应用程序的上下文参数，如图 12-40 所示。

单击“删除”按钮，可以删除选中的上下文参数。单击“添加”按钮或者“编辑”按钮，可以打开“添加 上下文参数”对话框，如图 12-41 所示。

其中，“参数名称”文本框用于指定参数的名称，如 name。“参数值”文本框用于指定参数值，如 Apple。“描述”文本框用于输入描述信息。

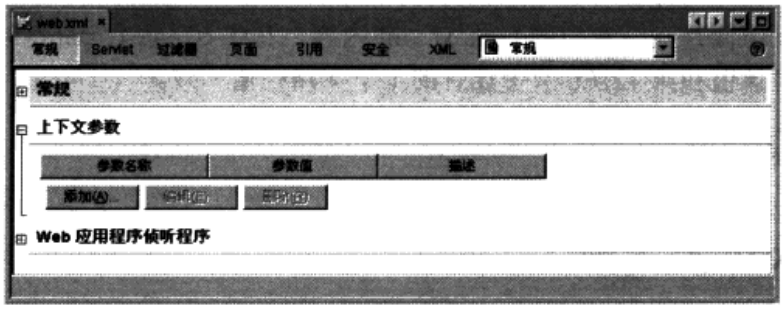


图 12-40 web.xml 描述符的上下文参数设置

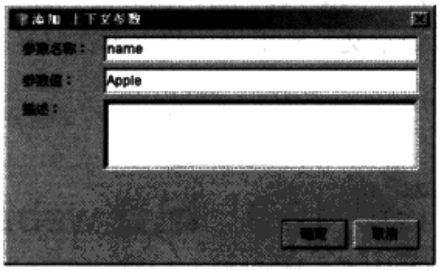


图 12-41 “添加 上下文参数”对话框

12.9.3 应用程序侦听程序

通过使用 Web 应用程序侦听程序，可以更好地控制与 ServletContext 和 HttpSession 对象之间的交互，并有效地管理 Web 应用程序使用的资源。Web 应用程序侦听程序是实现一个或多个 Servlet 事件侦听程序接口的类。

初始化和销毁 Servlet 上下文和会话时，以及在上下文或会话中添加或删除属性时，应用程序事件将可能通知 Web 应用程序侦听程序。

Download at http://www.pin5i.com/

在 web.xml 可视编辑器中选中“常规”标签，展开“上下文参数”节点，来添加、删除和查看 Web 应用程序侦听程序的部署信息，如图 12-42 所示。

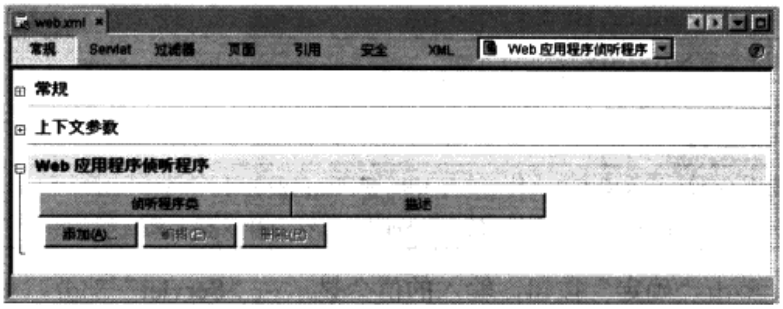


图 12-42 web.xml 描述符的应用程序侦听程设置

单击“删除”按钮，可以删除选中的应用程序侦听程序。单击“添加”按钮，或者“编辑”按钮，可以打开“添加 侦听程序”对话框，如图 12-43 所示。在“侦听程序类”文本框中，可以指定侦听程序的类，如 NewServletListener。单击旁边的“浏览”按钮，可以在项目中侦听程序类。在“描述”文本框中，可以输入侦听程序的描述信息。

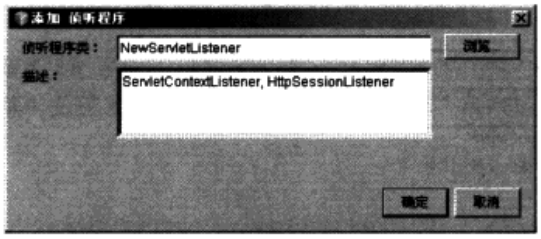


图 12-43 “添加 侦听程序”对话框



说明： *在 web.xml 文件中注册 Web 应用程序侦听程序前，请确认应用程序侦听程序已经存在。*



12.9.4 Servlet

要部署 Servlet，必须在 web.xml 文件中对其进行注册。通过执行此操作，可为服务器提供部署 Servlet 所需的信息。

在 web.xml 可视编辑器中选中“Servlet”标签，来添加、删除和查看 Web 应用程序的 Servlet 和 JSP 文件（可选）的部署信息。单击“添加 Servlet 元素”按钮，打开“添加 Servlet”对话框，如图 12-44 所示。对话框中各个参数的说明如表 12-3 所示。

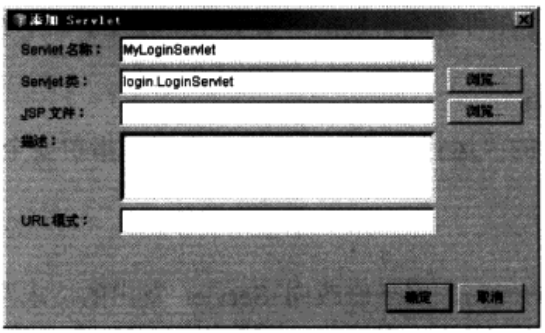


图 12-44 “添加 Servlet”对话框

Download at http://www.pin5i.com/

表 12-3 “添加 Servlet”对话框中各个参数的说明

参　数	描　述
Servlet 名称	用于设置 Servlet 或 JSP 文件的显示名称
Servlet 类	用于指定 Servlet 的类
JSP 文件	用于指定 JSP 文件
描述	用于输入描述信息
URL 模式	指定 Servlet 定义或 JSP 文件定义与 URL 模式之间的映射，如/myshop/*、*.jsp。对指定的 URL 模式的所有请求，由 Servlet 文件定义的关联 Servlet，或由 JSP 文件定义的关联 JSP 文件来处理

输入完成后，单击“确定”按钮，输入的值会显示在“Servlet”部分，如图 12-45 所示。

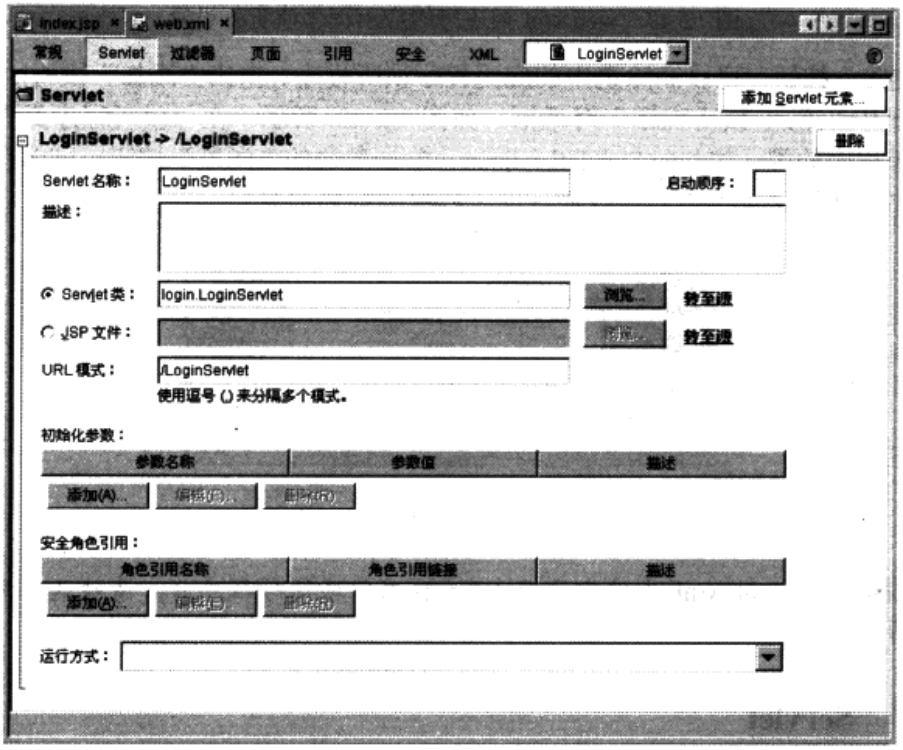


图 12-45　web.xml 描述符的 Servlet 设置

在“启动顺序”文本框中，可以输入一个整数，设置 Servlet 在应用程序中的启动顺序。如果要设置初始化参数，可以单击“初始化参数”部分的“添加”按钮。

单击“安全角色引用”部分的“添加”按钮，可以输入如下的内容。

- 角色引用名称：指定安全角色引用的显示名称。
- 角色引用链接：指定在可视编辑器的“安全性”部分中的“安全角色”表中定义的安全角色。
- 描述：指定描述信息。

“安全角色引用”部分的“运行方式”下拉列表框用于指定安全角色的运行方式。



12.9.5　过滤器

过滤器是一组可重用的代码，用于修改向 Servlet 发出的请求及其作出的响应。过滤器可以实现用户身份验证、日志和审计、本地化、数据压缩、XML 内容的 XSLT 变换等功能。

在 web.xml 可视编辑器中，选中“Servlet 过滤器”部分和“过滤器映射”部分来添加、删

Download at http://www.pin5i.com/

除和查看 Web 应用程序的 Servlet 过滤器的部署信息，如图 12-46 所示。其中各个参数的说明如表 12-4 所示。

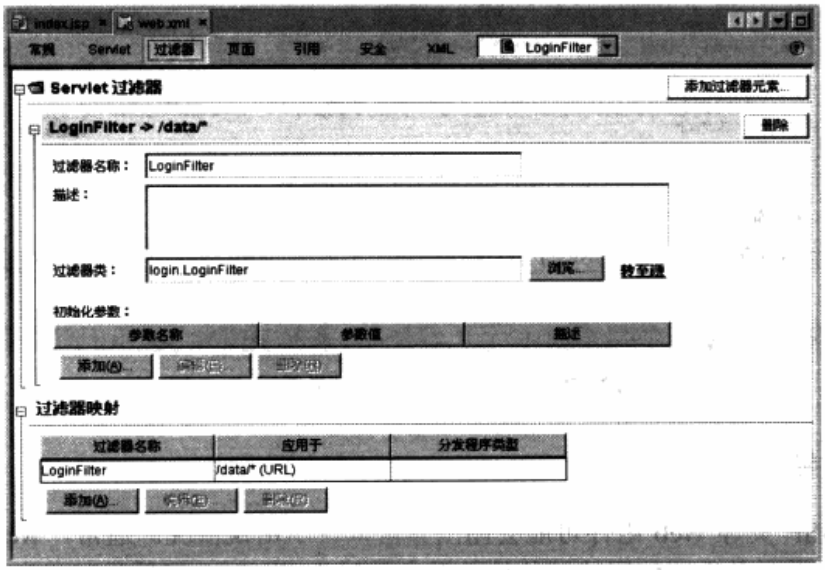


图 12-46　web.xml 描述符的过滤器设置

表 12-4　过滤器设置各个参数的说明

参　数	描　述
过滤器名称	指定过滤器的显示名称，如 NewSimpleFilter
过滤器类	指定过滤器的类，如 NewSimpleFilterClass
描述	指定描述信息
参数名称	定参数的显示名称
参数值	指定参数值
过滤器名称	选择映射要应用到的过滤器
应用于	指定过滤器要应用到的 URL 模式或 Servlet，如/mydata。
分发程序类型	指定过滤器应用于请求的方式，可以选择下面的一种或多种分发程序类型。 • 请求：默认方式，仅当请求直接来自客户端时为请求方式 • 前移：仅当请求已转发到组件时 • 包括：仅当请求将由某个被包含的组件处理时 • 错误：仅当请求将通过错误页面机制处理时



12.9.6　欢迎文件

欢迎文件是在部署 Web 应用程序时显示的 JSP 文件、HTML 文件或 Servlet。在 web.xml 可视编辑器中，选中“页面”标签，展开“欢迎文件”部分，来添加、删除和查看 Web 应用程序的欢迎文件，如图 12-47 所示。

单击“浏览”按钮，查找和选择所需的欢迎文件。可以选择多个欢迎文件，但只将第一个现有的欢迎文件用作 Web 应用程序的欢迎文件。可以将其他文件注册为欢迎文件进行测试和调试。

在开发阶段时，还可以通过第 12.7.6 节中介绍的方法，在项目属性的“相对 URL”文本框中键入欢迎文件。此时，会首先使用项目属性中的设置。只有当项目属性中的设置不存在时，

Download at http://www.pin5i.com/

才会使用 web.xml 文件中的设置。在项目部署到服务器后，只能使用 web.xml 中定义的欢迎文件。

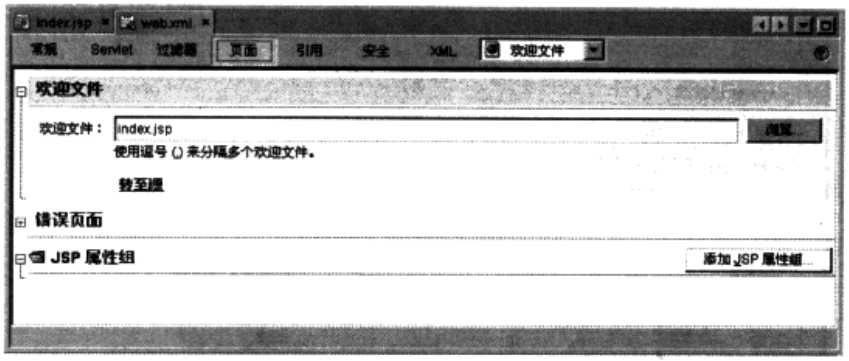


图 12-47 web.xml 描述符的欢迎文件设置

12.9.7 错误页面

Servlet 和 JSP 页等 Web 组件可能会抛出异常以说明错误情况。通常，Web 组件会捕获和处理其自己的异常以及它所使用的其他组件抛出的异常。当 Web 组件收到或生成它没有捕获的异常时，就会将该异常传递到 Web 容器。如果将另一个 Web 组件定义为该异常的错误页面，则请求会指向错误页面的 URL。

在 web.xml 可视编辑器中，选中“页面”标签，展开“错误页面”部分，来添加、删除和查看 Web 应用程序错误页面的部署信息，如图 12-48 所示。其中各个参数的说明如表 12-5 所示。

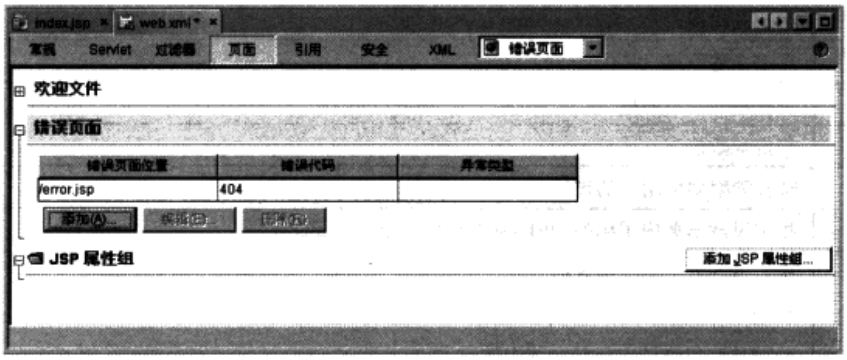


图 12-48 web.xml 描述符的错误页面设置

表 12-5 错误页面各个参数的说明

参 数	描 述
错误页面位置	指定用于处理错误的错误页面
错误代码	设置调用错误页面的 HTTP 错误代码，如 404。当设置了错误代码后，就不需要再设置异常类型了
异常类型	指定调用错误页面的 Java 异常类型，如 java.lang.NullPointerException。当设置了异常类型后，就不需要再设置错误代码了

12.9.8 JSP 属性组

JSP 属性组是为 Web 应用程序中的一组 JSP 文件定义的一组属性。在 web.xml 可视编辑器中，选中“页面”标签，展开“JSP 属性组”部分，来添加、删除和查看 Web 应用程序的 JSP 属性组，如图 12-49 所示。其中各个参数的说明如表 12-6 所示。

Download at http://www.pin5i.com/

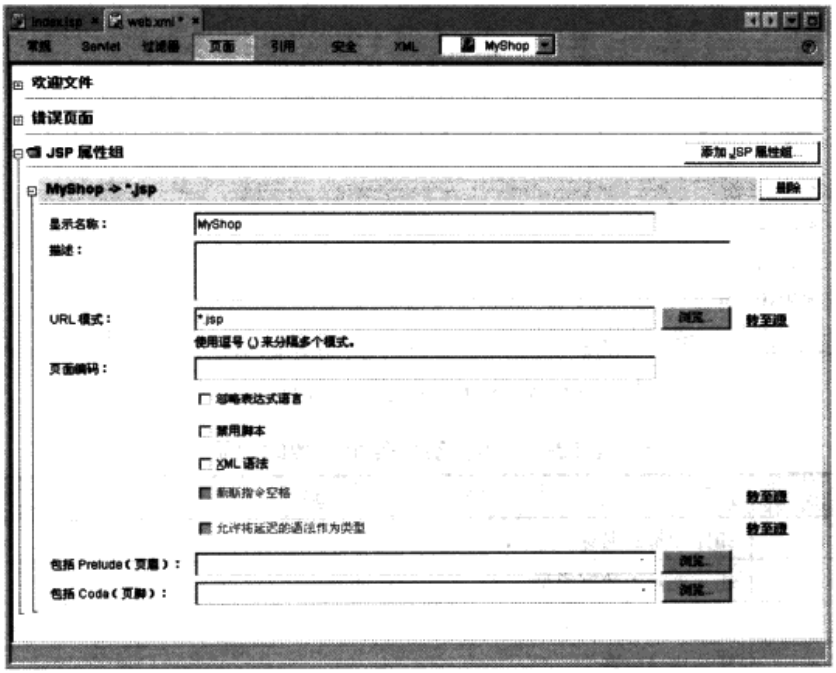


图 12-49　web.xml 描述符的 JSP 属性组设置

表 12-6　JSP 属性组各个参数的说明

参　数	描　述
显示名称	指定组的显示名称
描述	输入描述信息
URL 模式	指定将使用这些属性的 JSP 文件组，例如 myshop/*.jsp
页面编码	为组中所有 JSP 文件指定页面编码。有效值应与单个 JSP 文件的 page 指令的 pageEncoding 属性所使用的值相同。如果在 JSP 属性组中为页面编码定义某个值，然后在单个 JSP 文件的 pageEncoding 指令中为其指定另一个不同的值，则会出现转换时错误
忽略表达式语言	忽略属于该组的所有 JSP 文件的默认表达式语言（EL）计算模式。不同的 web.xml 版本，默认值也不相同。使用 Servlet 2.3 或更早版本的 web.xml 文件提供的 JSP 文件，其默认模式为忽略 EL 表达式，这会提供向后兼容性。Servlet 2.4 版本的 web.xml，其默认模式为计算 EL 表达式，这会自动提供大多数应用程序所需的默认值。未进行计算的 EL 表达式将作为一般文本进行处理
忽略 Java 脚本	使属于该组的所有 JSP 文件的脚本无效。默认情况下，JSP 文件中的脚本是有效的。在脚本无效时，如果组中的任何页包含 scriptlet、脚本表达式和声明，则它们会产生转换错误
XML 语法	指定组中的所有 JSP 页是 JSP 文档，而不是 JSP 文件。这意味着它们使用 JSP 文档语法，而不是标准 JSP 语法
包括 Prelude（页眉）	指定组中的所有 JSP 文件隐式包含页眉
包括 Coda（页脚）	指定组中的所有 JSP 文件隐式包含页脚



12.9.9　环境条目

应用程序环境条目是可以在整个 Web 应用程序中使用的初始化变量，例如可以使用它们来验证对象的类型。

在 web.xml 可视编辑器中，选中“引用”标签，展开“环境条目”部分，来添加、删除和查看 Web 应用程序的环境变量，如图 12-50 所示。其中各个参数的说明如表 12-7 所示。

Download at http://www.pin5i.com/

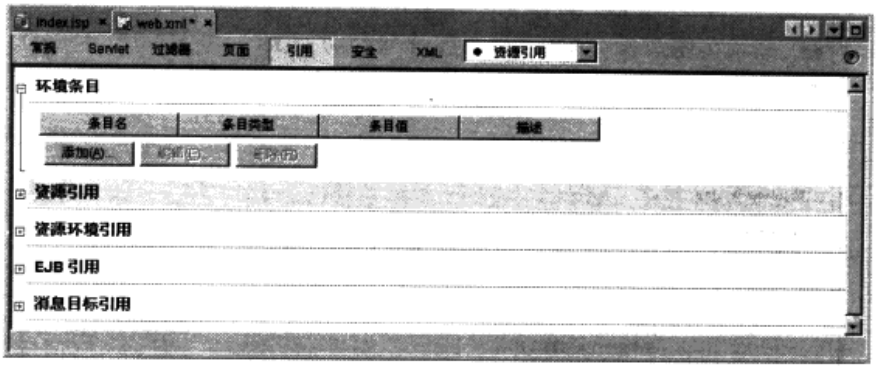


图 12-50 web.xml 描述符的环境条目设置

表 12-7 环境条目各个参数的说明

参 数	描 述
条目名	指定变量的名称
条目类型	指定变量的数据类型
条目值	条目变量对应的条目值
描述	指定描述信息



12.9.10 资源引用

要从 Web 应用程序中访问数据源、JMS 连接工厂、JavaMail 会话或 URL 链接，必须在该应用程序的 Web 应用程序部署描述符中声明资源引用。资源引用指定了在访问资源时使用的 JNDI 名称、对象类型和验证种类。

在 web.xml 可视编辑器中，选中“引用”标签，展开“资源引用”部分，来添加、删除和查看 Web 应用程序的资源引用，如图 12-51 所示。其中各个参数的说明如表 12-8 所示。

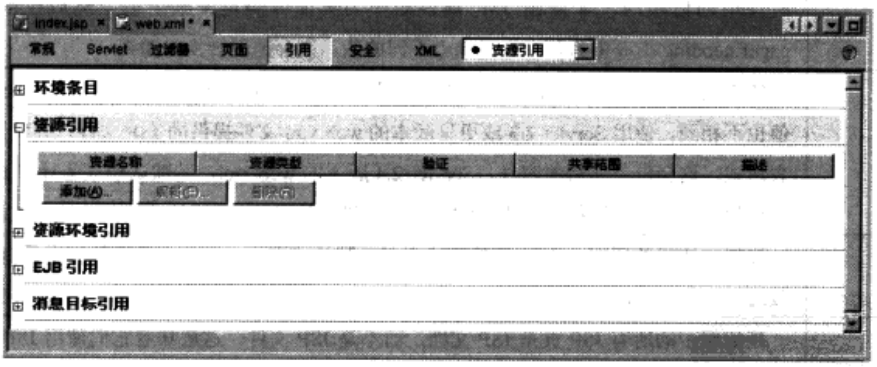


图 12-51 web.xml 描述符的资源引用设置

表 12-8 资源引用各个参数的说明

参 数	描 述
资源名称	指定资源引用的名称
资源类型	通过下拉列表框选择资源的类型，可以是如下的几个值。 • javax.sql.DataSource：数据源 • javax.jms.ConnectionFactory：连接工厂 • javax.mail.Session：会话 • java.net.URL：URL 链接

Download at http://www.pin5i.com/

续表

参　数	描　述
验证	指定验证凭证的来源，可以是如下的几个值。 • Container：默认值，由登录到服务器并创建组件实例的调用者提供 • Application：应用程序提供的编程验证
共享范围	指定是否共享资源，可以是如下的几个值。 • Shareable：进行资源共享 • Unshareable：不进行资源共享
描述	指定描述信息

12.9.11 资源环境引用

资源环境引用包含 Web 应用程序对管理对象（该对象与 Web 应用程序环境中的资源关联）的引用声明。它包含可选描述和资源环境引用名称，并且指明了 Web 应用程序代码需要的资源环境引用类型。

在 web.xml 可视编辑器中，选中“引用”标签，展开“资源环境引用”部分，来添加、删除和查看 Web 应用程序的资源环境引用，如图 12-52 所示。其中各个参数的说明如表 12-9 所示。

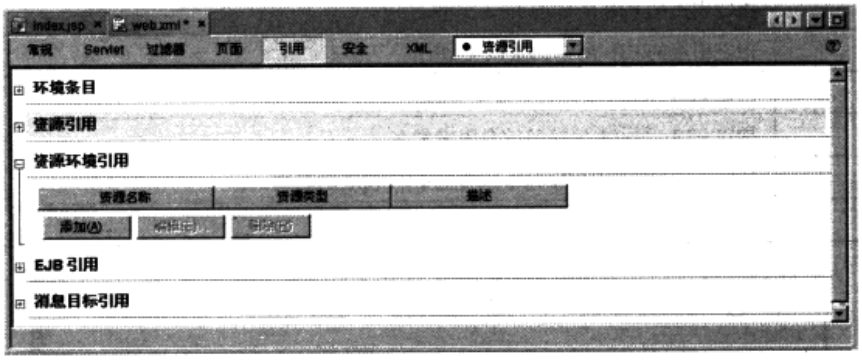


图 12-52　web.xml 描述符的资源环境引用设置

表 12-9　资源环境引用各个参数的说明

参　数	描　述
资源名称	指定资源环境引用的名称。指定的值是在 Web 应用程序代码中使用的环境条目名称。该名称是与 java:comp/env 上下文相对的 JNDI 名称，在 Web 应用程序中必须是唯一的
资源类型	指定资源环境引用的类型：javax.jms.Queue 或 javax.jms.Topic，是 Java 语言类或接口的全限定名称
描述	指定描述信息

12.9.12 EJB 引用

如果项目中使用了 EJB 组件，可以“EJB 引用”部分来添加、删除和查看 Web 应用程序的 EJB 模块引用，在部署时会将此处创建的引用映射到 EJB 模块的实际位置。

在 web.xml 可视编辑器中，选中“引用”标签，展开“EJB 引用”部分，如图 12-53 所示。其中各个参数的说明如表 12-10 所示。

12.9.13 消息目标引用

J2EE 1.4 规范允许 J2EE 组件使用名为消息目标引用的逻辑名称来引用消息目标对象。可

Download at http://www.pin5i.com/

以在 EJB 模块、应用程序客户端模块或 Web 模块中定义消息目标引用所指向的消息目标。

在 web.xml 可视编辑器中，选中“引用”标签，展开“消息目标引用”部分，来添加、删除和查看 Web 应用程序的消息目标引用，如图 12-54 所示。其中各个参数的说明如表 12-11 所示。

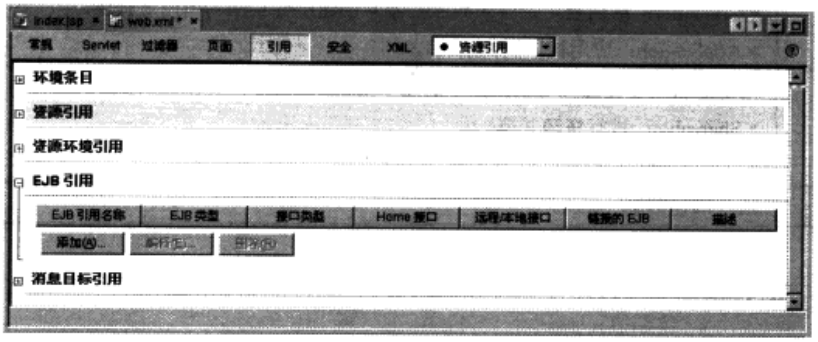


图 12-53　web.xml 描述符的 EJB 引用设置

表 12-10　EJB 引用各个参数的说明

参　数	描　述
EJB 引用名称	指定资源引用的名称
EJB 类型	指定引用的 EJB 所需的 Java 类型
接口类型	指定引用的 EJB 所需的接口类型
Home 接口	指定 EJB Home 接口的全限定类名
远程/本地接口	指定 EJB 远程/本地接口的全限定类名
链接的 EJB	指定 JavaEE 应用程序包中包含的 EJB 模块的名称
描述	指定描述信息

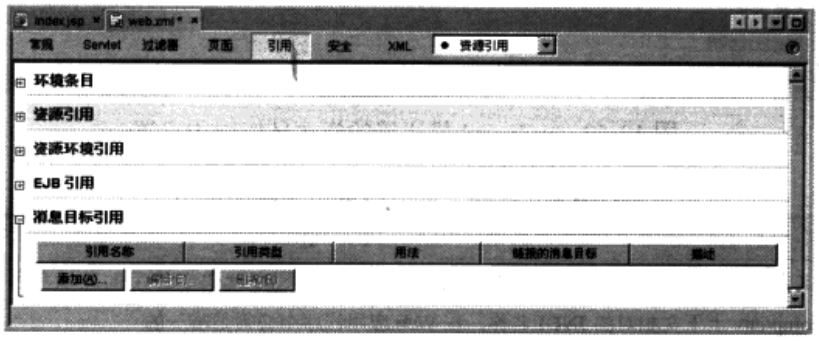


图 12-54　web.xml 描述符的消息目标引用设置

表 12-11　消息目标引用各个参数的说明

参　数	描　述
引用名称	指定引用的名称。消息目标引用的名称是相对于 java:comp/env context 的。如名称应该为 jms/SharePriceQueue，而不是 java:comp/env/jms/SharePriceQueue
引用类型	指定所需的引用目标类型：java.jms.Queue 或 java.jms.Topic
用法	指定消息目标如何处理消息。 • Produces：生产 • Consumes：消费 • ConsumesProduces：生产和消费两者皆有
链接的消息目标	消息目标的名称
描述	指定描述信息

Download at http://www.pin5i.com/

12.9.14 登录配置

在 web.xml 可视编辑器中，选中“安全”标签，展开“登录配置”部分，来配置 Web 应用程序的验证，如图 12-55 所示。其中各个参数的说明如表 12-12 所示。

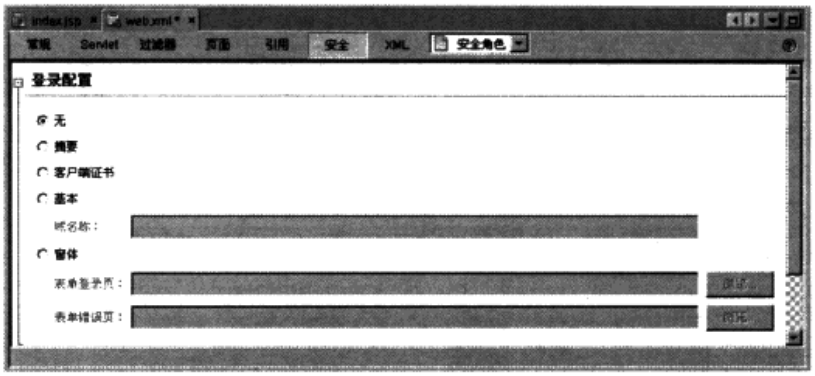


图 12-55 web.xml 描述符的登录配置设置

表 12-12 登录配置各个参数的说明

参 数	描 述
无	表示没有验证，这就相当于从部署描述符中删除了登录配置元素
摘要	一种验证机制。在该机制中，Web 应用程序通过向 Web 服务器发送消息摘要及其 HTTP 请求消息来对自身进行验证。通过对 HTTP 请求消息和客户端口令的组合使用单向散列算法，可以计算摘要。摘要通常比 HTTP 请求小得多，并且不包含口令
客户端证书	一种使用 HTTP over SSL 的验证机制。在该机制中，服务器和（可选）客户端通过公共密钥证书来彼此验证对方（该公共密钥证书符合由 X.509 公共密钥基础结构定义的标准）
基本	可以在“域名称”字段中键入安全域的名称，但无法对该值进行验证
窗体	允许为登录页和错误页选择 JSP 页

12.9.15 安全角色

在 web.xml 可视编辑器中，选中“安全”标签，展开“安全角色”部分，来添加、编辑和删除安全角色，如图 12-56 所示。其中各个参数的说明如表 12-13 所示。

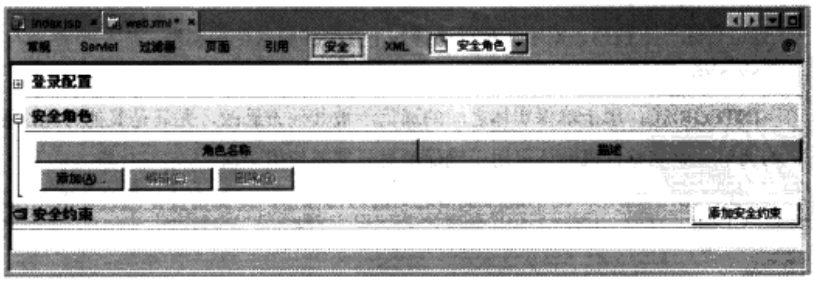


图 12-56 web.xml 描述符的安全角色设置

表 12-13 安全角色各个参数的说明

参 数	描 述
角色名称	安全角色的名称
描述	指定描述信息

Download at http://www.pin5i.com/

12.9.16 安全约束

在 web.xml 可视编辑器中，选中“安全”标签，展开“安全约束”部分，来添加、编辑和删除安全约束，如图 12-57 所示。其中各个参数的说明如表 12-14 所示。

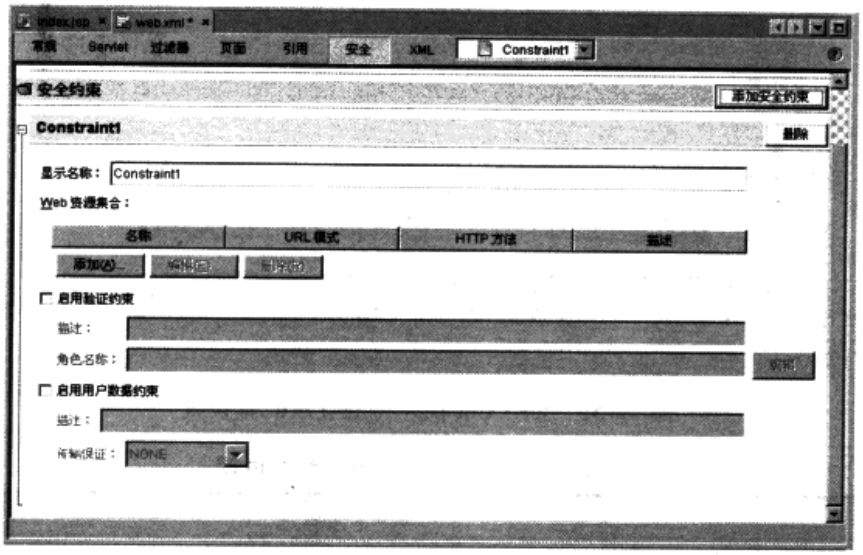


图 12-57 web.xml 描述符的安全约束设置

表 12-14 安全约束各个参数的说明

参数		描述
显示名称		标识安全约束的唯一名称
Web 资源集合		URL 模式和 HTTP 方法的列表，这些模式和方法描述了要保护的 Web 资源的集合
	名称	标识 Web 资源的唯一名称
	URL 模式	由逗号分隔的 URL 模式
	HTTP 方法	用于此 Web 资源的一种或多种 HTTP 方法
	描述	指定描述信息
启用验证约束		用于确定允许哪些特定角色访问 Web 资源集合的验证规则。选中该选项时，要在“角色名称”文本框中输入由逗号分隔的角色列表。这些角色必须与“安全角色”表中定义的角色相匹配
启用用户数据约束		指示应该如何保护客户端与 Web 容器之间的数据。此保护可以防止篡改数据或防止窃取数据。在“保证传输”下拉列表框中，可以选择如下几个传输保证的类型。 NONE：无任何的类型 INTEGRAL：用于确保实体之间的通信不被其他方篡改，尤其是其他方无法截取并修改实体之间的通信。此完整性机制还可用于确保消息只能被使用一次 CONFIDENTIAL：用于确保实体之间的通信保密。此保密性是通过对消息内容加密而获得的

12.10 配置 sun-web.xml 描述符

sun-web.xml 描述符用于配置 Sun Java System Application Server 的部署设置。在“项目”窗口中，展开 Web 应用程序项目的“配置文件”节点。双击 sun-web.xml，可以在 sun-web.xml 可视编辑器中将其打开并进行编辑。

Download at http://www.pin5i.com/

12.10.1 常规

在 sun-web.xml 可视编辑器中选择“常规”标签，可以在这里设置常规的属性，如图 12-58 所示。其中各个参数的含义如表 12-15 所示。

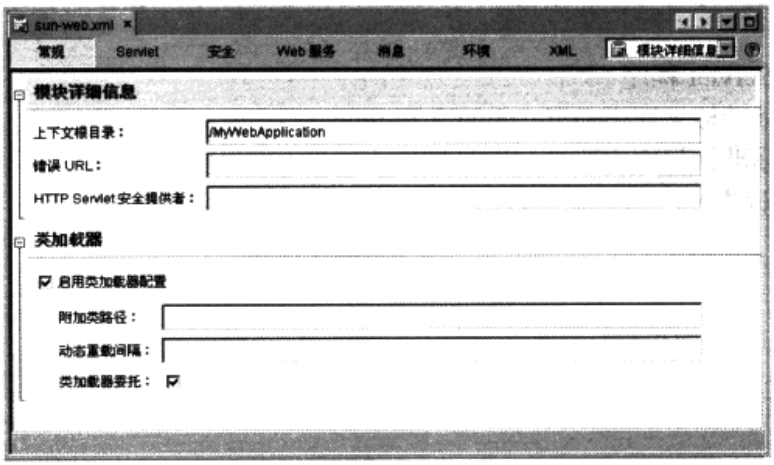


图 12-58 “常规”标签

表 12-15 “常规”标签中各个参数的说明

参 数	描 述
上下文根目录	上下文根目录是 WAR 文件的基目录。Servlet 上下文使用上下文根目录来解析 WAR 文件的路径
错误 URL	指定出现错误时重定向到的 URL
启用类加载器配置	选中该复选框，将启用类加载器
附加类路径	输入附加类的路径
动态重载间隔	设置动态重载的间隔时间

12.10.2 消息

在 sun-web.xml 可视编辑器中选择“消息”标签，提供对任何资源的引用，如图 12-63 所示。其中各个参数的含义如表 12-16 所示。

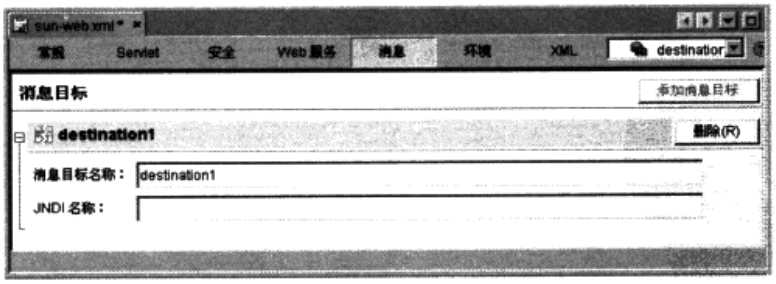


图 12-59 sun-web.xml 描述符的消息设置

表 12-16 “消息”标签中各个参数的说明

参 数	描 述
消息目标名称	显示逻辑消息目标的名称。消息目标名称必须与 web.xml 文件中相应的消息目标名称相匹配
JNDI 名称	显示消息目标的 JNDI 名称

Download at http://www.pin5i.com/

12.11 小结

本章主要介绍 Web 应用程序开发中的一些高级技术，例如如何使用标记、如何进行侦听等。本章的学习重点是如何使用标准标记库，以及如何自定义标记。通过本章的学习，读者应该掌握以下基本内容：

- 如何开发 Servlet 程序
- 如何使用标准标记库中的 if、choose 和 foreach 标记
- 如何使用标准标记库中的数据库处理库
- 配置 Web 应用程序部署描述符

Download at http://www.pin5i.com/

第13章

在NetBeans中开发Struts程序

Struts项目是由Apache Software Foundation发起的开源项目，它在Java服务器端实现了MVC（Model-View-Controller）设计模式。Struts为使用MVC体系编写应用提供了一个架构，因而目前非常流行。

为了能让读者在学习NetBeans开发Struts时，不必再找另外的书学习Struts，而影响学习的连贯性。本章的第一节先对MVC设计模式、Struts框架结构等知识进行必要的介绍。从第二节开始，将以一个登录模块为例，介绍如何在NetBeans中快速开发Struts应用程序。

本章主要内容

- MVC设计模型
- Web应用程序中的MVC模型
- 如何在NetBeans中创建Struts ActionForm Bean
- 如何在NetBeans中创建Struts Action

本章亮点

- 为项目增加Strtus框架支持
- Struts框架工作原理

13.1 Struts概述

Struts是一个开源的应用框架，在最近几年得到了飞速的发展，在JSP Web应用开发中应用得非常广泛。Struts框架的实质就是JSP MVC Model 2模型的实现。下面将首先讨论MVC设计模式，然后在此基础上介绍JSP的两个MVC模型以及Struts框架结构。

13.1.1 MVC设计模式

MVC是一种被广泛采用的设计模式，Struts框架也是采用了MVC模式，因此有必要对MVC模式进行一个简单的介绍。

MVC最初是在Smalltalk-80中被用来构建用户界面的。MVC是Model-View-Controller的缩写，M代表模型Model；V代表视图View；C代表控制器Controller。

MVC的目的是增加代码的重用率，减少数据表达、数据描述和应用操作的耦合度。通过把数据模式从各种可以被存取和控制的数据中分离出来可以改善分布式系统的设计。MVC设计模式由三部分组成，如图13-1所示。

模型是应用对象，没有用户界面，是应用程序的主体部分。模型主要用于表示业务数据和业务逻辑，其具体的功能如下所示：

- 抽象系统的功能，提供使用系统功能的方法和路径；
- 封装应用程序状态；

Download at http://www.pin5i.com/

- 管理数据的存储和一致性；
- 处理业务逻辑；
- 通知视图的更新。

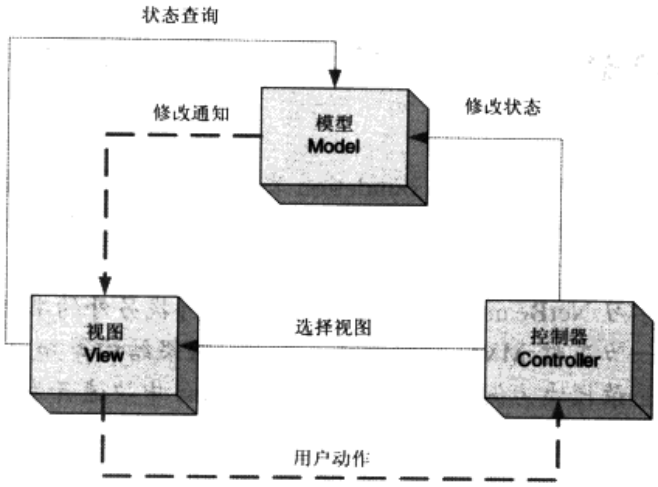


图 13-1 MVC 设计模式

视图是应用程序中用户界面相关的部分，视图向用户显示数据，并能接收用户的输入数据，但它并不进行任何实际的业务处理。视图的主要功能为：

- 向控制器发送查询请求；
- 显示数据；
- 接收来自模型的更新。

控制器定义用户界面对用户输入的响应方式，负责把用户的动作转成针对 Model 的操作。Model 在进行输入数据的处理后再更新视图，把处理的响应反馈给用户。控制器的主要功能为：

- 定义应用程序的行为；
- 映射用户请求到模型；
- 根据用户的输入和上下文情况选择合适的显示数据；
- 选择合适的视图。

13.1.2 Web 应用程序的 MVC 实现模式

在 Web 应用程序中，存在着很多种编程模型（也被称为实现模式）。图 13-2 是一个典型的 JSP 应用程序编程模型，上一章中基于 JSP 技术的登录模块程序就是采用这个编程模型实现的。

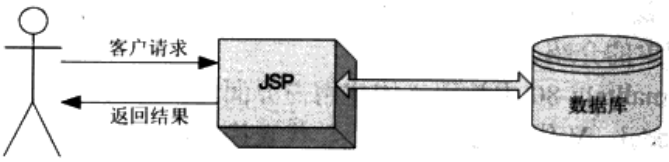


图 13-2 典型的 JSP 应用程序编程模型

JSP 文件负责业务逻辑、控制网页流程、访问数据库。当用户发出请求后，JSP 文件会处理用户的请求并根据需要访问数据库，最后把返回的数据发送给用户。这个模型的好处是比较简单，用户只要在 JSP 文件中编写代码就可完成所有的处理逻辑。其缺点是给 Web 开发带来了强耦合、调试困难、程序处理逻辑复杂等问题。

为了解决这些问题，Sun 公司又提出了另外一种利用 JavaBean 的编程模型，如图 13-3 所示。上一章中使用 JSP + JavaBean 技术的登录模块程序就是采用这个编程模型实现的。

Download at http://www.pin5i.com/

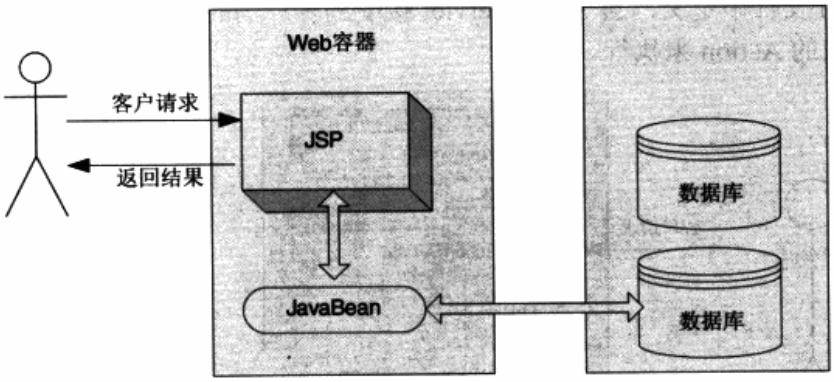


图 13-3 使用 JavaBean 的编程模型

这是比较经典的一种编程模型，属于 MVC 的一种具体实现方式。在这个模型中，JSP 充当着控制器和视图的作用，JavaBean 为模型。在客户发送请求后，JSP 处理客户请求，然后调用模型进行业务处理。模型返回数据后，再由 JSP 返回给客户显示。这个模型是由图 13-2 中的模型发展而来的，因而实现起来比较容易。但是，由于 JSP 页面中既包含了 HTML 代码，又包含了 Java 语言，因此非常不利于程序的维护。

为了解决这个模型中视图和控制器耦合的缺点，Sun 公司又提出了利用 Servlet 作为控制器的编程模型，如图 13-4 所示。上一章中使用 JSP + JavaBean + Servlet 技术的登录模块程序就是采用这个编程模型实现的。

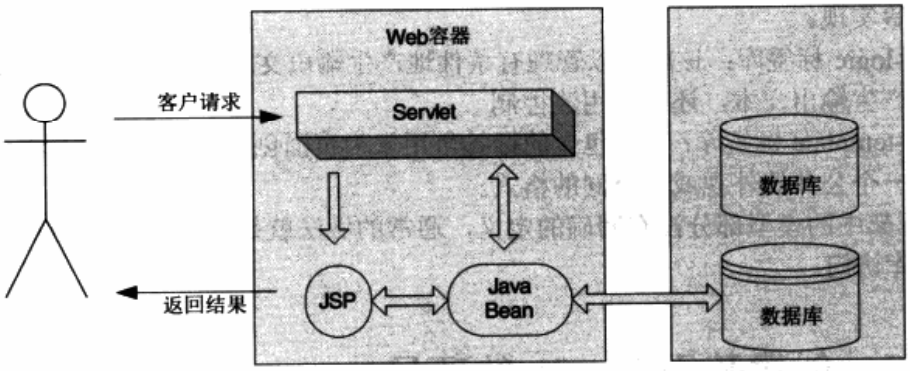


图 13-4 利用 Servlet 作为控制器的编程模型

这个模型中使用 Servlet 作为控制器，JSP 作为视图只单纯地负责显示，JavaBean 为模型。当用户发送请求后，Servlet 会根据用户的请求调用 JavaBean 完成业务处理，然后 JavaBean 将处理结果返回，并通过 JSP 发送给客户进行显示。这个模型很好地区分了控制器、视图和模型，符合 MVC 的设计思想，因而对于大型系统的开发很有帮助。

13.1.3 Struts 框架结构及原理

Struts 是 Apache Jakarta 项目的一部分，为创建 Java Web 应用提供了一个开源框架。它的目标是实现 MVC 编程模型，即希望分离 Web 程序的表示层、控制层和后台功能层。Struts 框架实质就是利用 Servlet 作为控制器的编程模型的实现，如图 13-5 所示。

从图中可以看出，Struts 框架中使用 ActionServlet 和 Action 替代了图 13-4 中的 Servlet 和 JavaBean。ActionServlet 类是 Struts 框架的核心组件，它继承了 HttpServlet 类。ActionServlet 在 MVC 模型中扮演中央控制器的角色，客户端的请求都会由 ActionServlet 进行处理。Action 为子控制器，执行具体的控制器功能，并调用相应的后台模型来完成业务逻辑。Action 在

Download at http://www.pin5i.com/

Struts-config.xml 文件中定义，当 ActionServlet 接受到客户端的请求后，会到 Struts-config.xml 文件中查找相应的 Action 来执行。

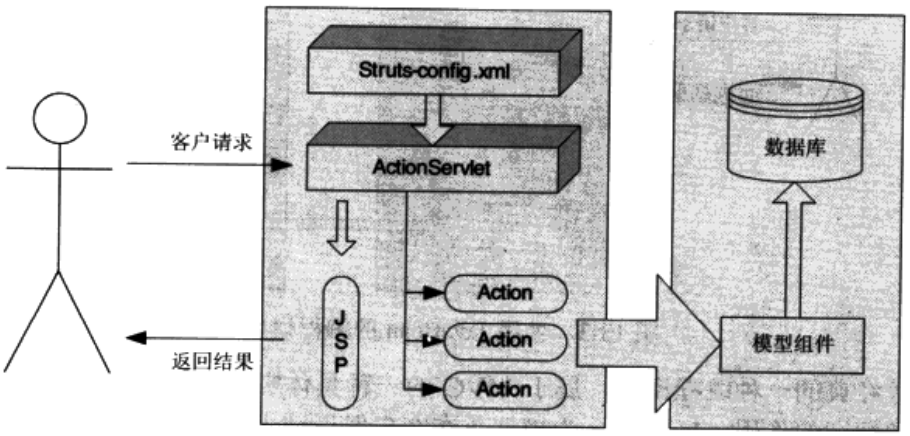


图 13-5 Struts 框架组件模型

视图是通过一组 JSP 程序实现的。除了基本的 JSTL 标记外，Struts 还定义了大量的标记来支持用户的开发。

- struts-html 标签库：该库可以用作创建动态 HTML 用户界面或者表格。例如，ActionForm 标记用于提交数据。
- struts-bean 标签库：对于将 JavaBean 代码放到 JSP 页面，该库提供了一个加强方式，通过标签实现。
- struts-logic 标签库：该库可以管理有条件地产生输出文本，遍历对象 collections 可以重复地产生输出文本，还有应用流控制。
- struts-template 标签库：该库包含的标签可用来为页面创建动态的 JSP 模板，这些页面都拥有一个公共的外观或者公同的格式。

Struts 框架中的模型部分没有明确的定义，通常的做法就是采用 EJB、Hibernate 等作为模型进行数据库访问。

13.2 创建基于 Struts 的项目

本节就以前面反复采用的登录模块为例，利用 Struts 框架实现这个功能。首先创建一个名为 StrutsDemo 的 Web 应用程序。创建项目时，在如图 13-6 所示的对话框中，选中“Struts 1.2.9”复选框，为项目增加 Struts 框架的支持。

在“操作 URL 模式”下拉列表框中，可以选择操作 URL 的模式。在“应用程序资源”文本框中，可以输入程序使用的资源。选中“添加 Struts TLD”复选框，会为项目添加 TLD。

说明：对于已经创建的项目，可以在“项目”视图中，右键单击项目节点，选择弹出菜单中的“属性”命令，打开项目属性对话框。在左边的“类别”列表中选择“库”节点，然后在右边的编译时库列表中添加 Struts 框架库。

13.3 使用 Struts 标记创建 JSP 文件

在 Struts 框架应用程序中，可以使用 Struts 提供的标记来编写 JSP 文件（视图）。在创建

Download at http://www.pin5i.com/

StrutsDemo 项目后，删除自动创建的 welcomeStruts.jsp 文件，然后从以前的 JSP_JavaBean_Servlet_Login 登录模块中复制 index.jsp、error.jsp、welcome.jsp 文件和 images 目录到当前项目中。把 JSP_JavaBean_Servlet_Login 项目中 src/java 目录下的 login 子目录复制到当前项目的对应位置。修改 index.jsp 文件如下所示，使用 Struts 标记代替 HTML 的表单。

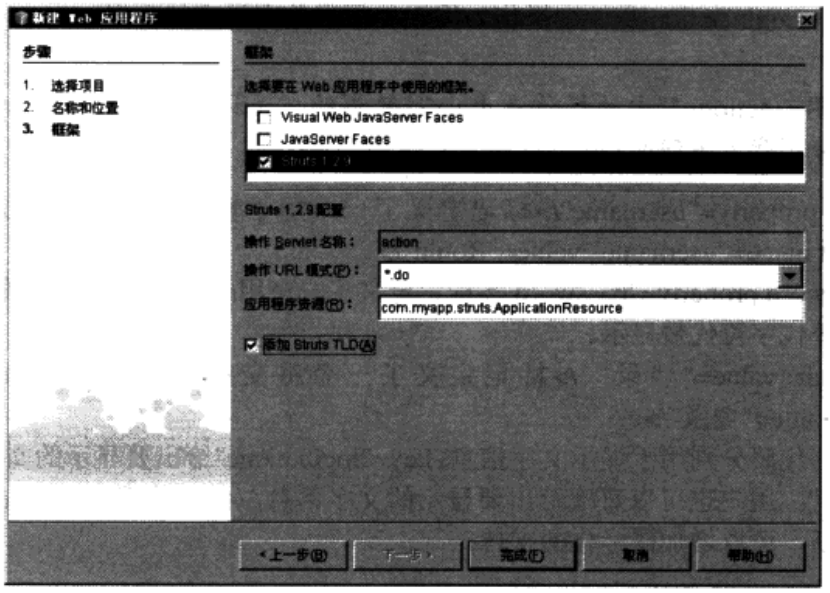


图 13-6 创建 Web 项目时添加 Struts 框架的支持

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean" prefix="bean" %>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-html" prefix="html" %>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>登录</title>
    </head>
    <body>

        <h1>请输入登录系统的用户名和密码</h1>
        <p></p>

        <html:form action="LoginAction" method="POST">
            <table border="1">
                <tbody>
                    <tr>
                        <td><bean:message key="login.name"/></td>
                        <td><html:text property="username"/></td>
                    </tr>
                    <tr>
                        <td><bean:message key="login.password"/></td>
                        <td><html:password property="password"/></td>
                    </tr>
                </tbody>
            </table>
            <br>
            <html:submit value="登录" />
            <html:errors />
        </html:form>
    </body>
</html>
```

Download at http://www.pin5i.com/

其中，下面的两行代码应用了 Struts 框架中的两个标记库。

```
<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean" prefix="bean" %>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-html" prefix="html" %>
```

<html:form>标记可以代替 HTML 的表单。action="LoginAction"属性指定了提交表单后的处理代码。这里 LoginAction 是一个操作，用于处理表单中提交的数据。



说明：action=""中的操作是大小写敏感的，action="LoginAction"和 action="loginAction"将分别代表不同的操作。

<html:text property="username"/>标记定义了一个用于输入用户姓名的文本框，相对于 HTML 的<input name="username" value="Robin"/>。

<html:password property="password"/>标记定义了一个用于输入用户密码的文本框，用户输入的密码将被替代字符代替显示。

<html:submit value="登录" />标记定义了一个提交按钮，相对于 HTML 的<input type="submit" value="登录"/>。

下面的两行代码分别用于显示文字信息，key="login.name"给出要显示的文字信息的关键字为“login.name”。用户也可以直接给出要显示的文字信息，但是使用<bean:message>标记将使得程序更加灵活，可以增加国际化的支持。

```
<bean:message key="login.name"/>
<bean:message key="login.password"/>
```

在“项目”视图中选择“源包”→“com.my.struts”→“ApplicationResource”文件，在 index.jsp 文件中显示的文字信息在这里进行定义。在文件的最后添加如下的资源。

```
login.info=Login Form
login.name=UserName
login.password=Password
error.name.required=Enter a name!
```

13.4 创建 ActionForm Bean

在视图中使用 Struts 创建表单后，就要创建 ActionForm 来保存表单提交的数据。选择“文件”→“新建文件”命令，打开“新建文件”对话框。在对话框左边的“类别”列表中选择“Struts”节点，在右边的“文件类型”列表中选择“Struts ActionForm Bean”节点，创建新的窗体 Bean 类，该类扩展于 Struts ActionForm 类。

单击“下一步”按钮，打开“新建 Struts ActionForm Bean”对话框，如图 13-7 所示。在“类名”文本框中输入 ActionForm 的类名字，这里输入 LoginActionForm。在“包”下拉列表框中选择源代码包 com.myapp.struts。

在“超类”下拉列表框中选择新建类的超类，这里选择 org.apache.struts.action.ActionForm。在“配置文件”下拉列表框中选择使用的配置文件，这里使用 Struts 默认的配置文件/WEB-INF/struts-config.xml。

单击“完成”按钮，创建 ActionForm 类，然后修改代码如下所示。

```
package com.myapp.struts;
/*
 * LoginActionForm.java
 *
 * Created on 2007年3月29日, 下午12:42
```

Download at http://www.pin5i.com/

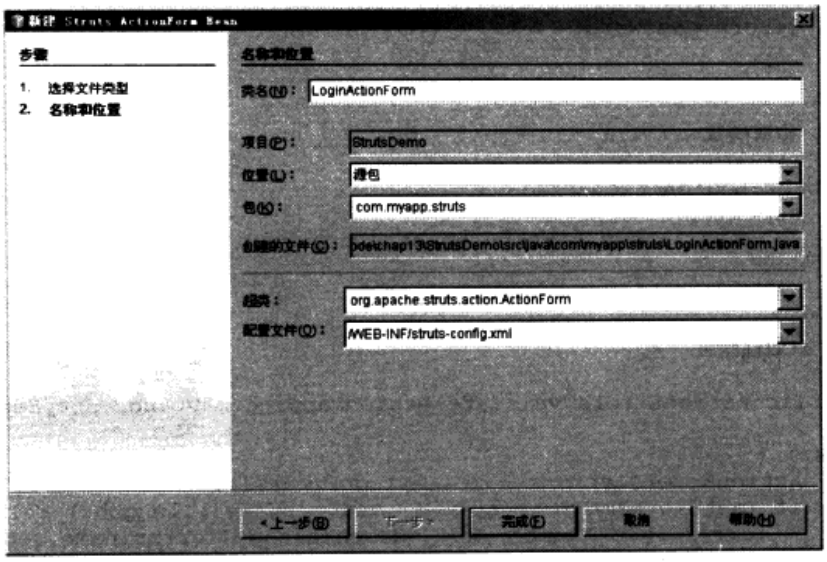


图 13-7 “新建 Struts ActionForm Bean”对话框

```
*/

import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

/**
 *
 * @author Liu Bin
 * @version
 */

public class LoginActionForm extends org.apache.struts.action.ActionForm {
    /** 登录用户名 */
    private String username;
    /** 用户密码 */
    private String password;

    /**
     * @return 登录用户名
     */
    public String getUsername() {
        return username;
    }

    /**
     * @param name 登录用户名
     */
    public void setUsername(String name) {
        username = name;
    }

    /**
     * @param password 用户密码
     */
    public void setPassword(String password) {
        this.password = password;
    }

    /**
     * @param password 用户密码
     */
```

Download at http://www.pin5i.com/


```
    public String getPassword() {
        return password;
    }

    /**
     * 构造函数
     */
    public LoginActionForm() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * 有效性验证
     */
    public ActionErrors validate(ActionMapping mapping, HttpServletRequest

request) {
        ActionErrors errors = new ActionErrors();
        if (getUsername() == null || getUsername().length() < 1) {
            errors.add("name", new ActionMessage("error.name.required"));
            // TODO: add 'error.username.required' key to your resources
        }
        return errors;
    }
}
```

在这里，定义了两个成员变量：username 和 password。这两个变量的定义要和 index.jsp 文件中表单组件的名字一致。

创建 FormAction Bean 后，会在 struts-config.xml 文件中自动添加如下的代码，对 LoginActionForm 进行注册。

```
<form-beans>
  <form-bean name="LoginActionForm" type="com.myapp.struts.LoginActionForm"/>

</form-beans>
```

13.5 创建 Action 操作

在创建了 ActionForm Bean 后，需要创建 Action 操作，对表单进行处理。选择“文件”→“新建文件”命令，打开“新建文件”对话框。在对话框左边的“类别”列表中选择“Struts”节点，在“文件类型”列表中选择“Struts 操作”节点，创建新的操作类，该类扩展于 Action 类。

单击“下一步”按钮，打开“新建 Struts 操作”对话框，如图 13-8 所示。在“类名”文本框中输入操作的类名字，这里输入 LoginAction。

在“包”下拉列表框中选择源代码包 com.myapp.struts。在“超类”下拉列表框中选择新建类的超类，这里选择 org.apache.struts.action.Action。在“配置文件”下拉列表框中选择使用的配置文件，这里使用 Struts 默认的配置文件/WEB-INF/struts-config.xml。在“操作路径”文本框输入操作的路径，这里输入“/LoginAction”。

单击“下一步”按钮，打开“ActionForm Bean 和参数”对话框，如图 13-9 所示。选中“Action Form Bean”复选框，指定要与当前操作相关联的 ActionForm Bean。在“ActionForm Bean 名称”下拉列表框中选择要关联的 ActionForm Bean 的名字，这里选择刚刚创建的 LoginActionForm。

选择“输入资源”单选按钮，然后可以在后面的文本框中指定与 ActionForm Bean 关联的资源。

选择“输入操作”单选按钮，然后可以在后面的下拉列表框中选择与 ActionForm Bean 关联的操作。

Download at http://www.pin5i.com/

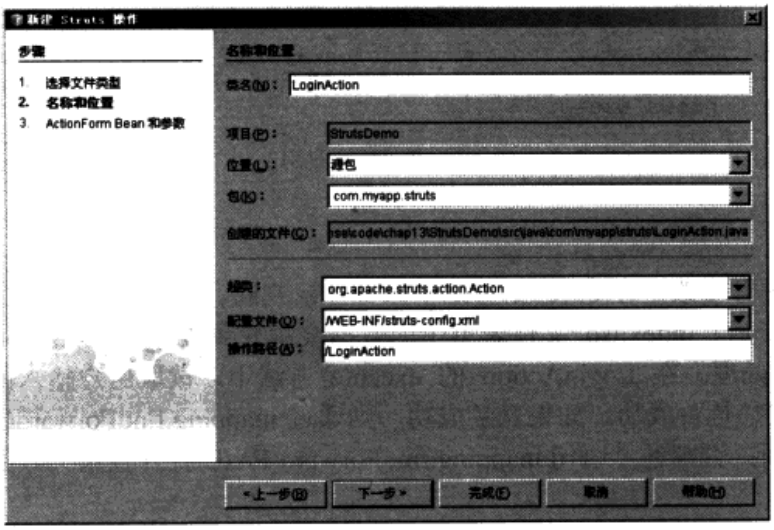


图 13-8 “新建 Struts 操作”对话框

选中“会话”单选按钮，指定 ActionForm Bean 的有效范围为会话范围；选中“请求”单选按钮，指定 ActionForm Bean 的有效范围为整个请求范围。

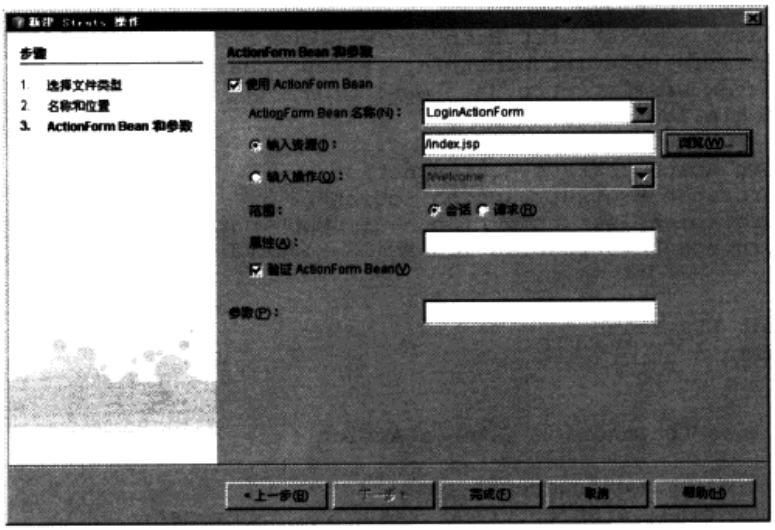


图 13-9 “ActionForm Bean 和参数”对话框

在“属性”文本框中可以指定可选属性。选中“验证 ActionForm Bean”复选框，指定对 ActionForm Bean 进行检验。在“参数”文本框中，可以指定可选参数。

单击“完成”按钮，自动创建如下的代码。其中，execute 方法用于执行具体的操作，mapping.findForward(SUCCESS)用于把页面重定向到另外的页面。

```
public class LoginAction extends Action {

    /* forward name="success" path="" */
    private final static String SUCCESS = "success";

    /**
     * This is the action called from the Struts framework.
     * @param mapping The ActionMapping used to select this instance.
     * @param form The optional ActionForm bean for this request.
     * @param request The HTTP Request we are processing.
     * @param response The HTTP Response we are processing.
     * @throws java.lang.Exception
```

Download at http://www.pin5i.com/


```
     * @return
     */
    public ActionForward execute(ActionMapping mapping, ActionForm  form,
            HttpServletRequest request, HttpServletResponse response)
            throws Exception {

        return mapping.findForward(SUCCESS);

    }
}
```

对以上代码进行修改，在 execute 方法中添加登录判断。当用户在 index.jsp 页面中输入用户名和密码并提交后，index.jsp 文件表单中的数据会填充到 LoginActionForm 中，并转交给 LoginAction 进行处理。在 LoginAction 的 execute 方法中，读取用户输入的数据，然后通过 LoginBean 判断登录是否成功。如果登录成功，则通过 mapping.findForward(SUCCESS)语句打开登录成功的页面；否则通过语句 mapping.findForward(FAIL)打开登录失败的页面。

```
/*
 * LoginAction.java
 *
 * Created on 2007年3月29日, 下午1:42
 */

package com.myapp.struts;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import login.LoginBean;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionForward;
/**
 *
 * @author Liu Bin
 * @version
 */

public class LoginAction extends Action {

    /* forward name="success" path="" */
    private final static String SUCCESS = "success";
    /* forward name="success" path="" */
    private final static String FAIL = "fail";

    /**
     * This is the action called from the Struts framework.
     * @param mapping The ActionMapping used to select this instance.
     * @param form The optional ActionForm bean for this request.
     * @param request The HTTP Request we are processing.
     * @param response The HTTP Response we are processing.
     * @throws java.lang.Exception
     * @return
     */
    public ActionForward execute(ActionMapping mapping, ActionForm  form,
            HttpServletRequest request, HttpServletResponse response)
            throws Exception {

        LoginActionForm loginActionForm = (LoginActionForm) form;

        boolean loginsuccess = false;

        //以下代码利用 LoginBean 进行登录判断
```

Download at http://www.pin5i.com/


```
            boolean loginflag = false;
            LoginBean bean = new LoginBean();
            bean.setUsername(loginActionForm.getUsername());
            bean.setPassword(loginActionForm.getPassword());
            loginsuccess = bean.checklogin();

            if (loginsuccess) {
                HttpSession session = request.getSession(true);
                session.setAttribute("username", loginActionForm.getUsername());
                return mapping.findForward(SUCCESS);
            } else {
                return mapping.findForward(FAIL);
            }
        }
    }
```

创建 LoginAction 类后，向导会自动在 struts-config.xml 文件中添加如下的代码，注册 Struts 操作。

```
    <action-mappings>
        <action input="/index.jsp" name="LoginActionForm"
                path="/LoginAction" scope="session"
                type="com.myapp.struts.LoginAction"/>
    </action-mappings>
```

这里，请删除如下的代码，因为这些代码是关于删除的 welcomeStruts.jsp 文件的，这里不需要这些内容。

```
    <global-forwards>
        <forward name="welcome"  path="/index.jsp"/>
    </global-forwards>

<action path="/Welcome" forward="/welcomeStruts.jsp"/>
```

在<action></action>标记中，输入“<”会显示如图 13-10 所示的提示框。这里选择“forward”。

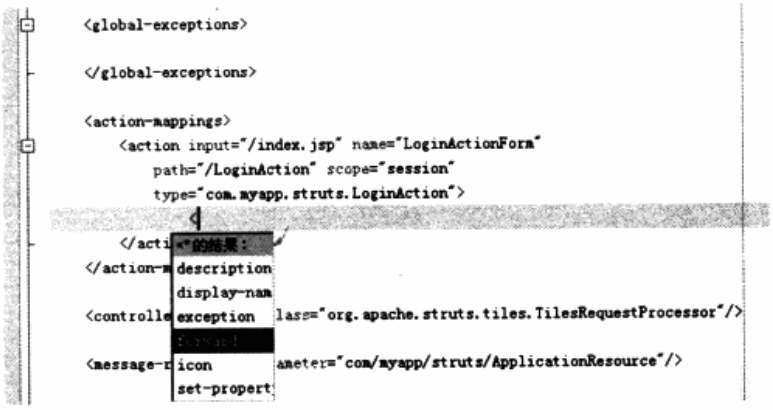


图 13-10　NetBeans 提供了对 struts-config.xml 编辑的支持

在<action></action>标记中添加如下的两行代码，分别对应用于 LoginAction 类的 execute 方法的 mapping.findForward(SUCCESS)和 mapping.findForward(FAIL)语句。

```
    <action input="/index.jsp" name="LoginActionForm"
            path="/LoginAction" scope="session"
            type="com.myapp.struts.LoginAction">
        <forward name="success" path="/welcome.jsp"/>
        <forward name="fail" path="/error.jsp"/>
    </action>
```

其中，name 属性的名字应该和 LoginAction 类中定义的如下两个常量一致：

```
/* forward name="success" path="" */
```

Download at http://www.pin5i.com/


```
private final static String SUCCESS = "success";
/* forward name="success" path="" */
private final static String FAIL = "fail";
```

至此，完成了一个简单的基于 Strtus 框架的登录模块，在项目中添加了 Java DB 数据库驱动程序库后，就可以运行该项目了。项目的运行结果与以前的一致，这里不再给出，说明登录模块运行正常。

13.6 小结

登录模块是一个简单而且常用的系统模块。本章以这个模块的开发为例，这样读者比较容易理解。相信读者在学习完本章以后，已经对 Struts 应用程序的开发有了一个大致的了解。鉴于本书主要是介绍 NetBeans 工具的使用方法，因此不能对 Struts 程序的开发作更深入的讲解。本章的内容安排能够起到抛砖引玉的作用。通过本章的学习，读者应该掌握如下的内容：

- Struts 框架的工作原理
- 如何在 NetBeans 中创建 ActionForm 以及 Action
- 如何编写 struts-config.xml 文件

Download at http://www.pin5i.com/

第 14 章

进行可视化的 Web 开发

在前面章节中介绍了一般的 Web 应用程序的开发，以及 Struts 框架的使用方法。本章将介绍目前比较流行的一种可视化的开发技术，即使用 JSF（JavaServer Faces）组件进行可视化的 Web 开发。

JSF 是业界流行的一种开发 Web 应用程序的框架，利用该框架可以像设计 Windows 应用程序一样，在 Web 页面中添加组件并处理用户事件。说得形象一些，利用 JSF 技术，用户可以像开发 ASP.NET 应用程序一样进行基于 Java 的 Web 应用程序开发。

本章主要内容

- 安装可视化的 Web 应用程序开发包
- 创建可视化的 Web 应用程序
- 进行页面导航
- 进行 AJAX 开发

本章亮点

- 可视化的页面编辑器
- 如何在页面中添加 AJAX 组件
- 使用 JSF 实现登录模块

14.1 可视化的 Web 开发概述

在上两章中介绍了开发 Web 应用程序的一般办法，本章介绍以可视化的方法开发 Web 应用程序。进行可视化的 Web 开发，需要安装 NetBeans Visual Web Pack 开发包。在 NetBeans Visual Web Pack 开发包提供了业界领先的 JavaServer Faces 组件支持，开发人员可以拖放 JavaServer Faces 组件来设置属性，在服务器端事件处理程序中编写代码。本节首先对 JavaServer Faces 组件进行概括介绍，然后介绍 Visual Web Pack 开发包的安装方法。

14.1.1 JavaServer Faces 开发概述

JavaServer Faces（JSF）是一种用于构建 Web 应用程序的新标准 Java 框架，它提供了一种以组件为中心，以可视化的方式来开发 Java Web 用户界面的方法。JavaServer Faces 技术规范为 JCP（Java Community Process）的 JSR-127 规范，提供了一种标准化的技术规范。

同其他可视化的 Web 开发技术（例如 ASP.NET 2.0）类似，JSF 提供了可视化的 UI 组件，使开发人员能够使用预先构建的用户界面组件，以可视化的方式来构建 Web 用户界面，从而提高了开发效率。

JavaServer Faces 包括以下两个组件库。

- UI 组件库：是一个定制标签库，用于在界面上显示 UI 组件。UI 组件库又分为如表 14-1 所示的几个部分。

Download at http://www.pin5i.com/

- 核心库：辅助常见的应用程序开发任务，例如国际化、事件处理、验证、转换输入数据等。

表 14-1 UI 组件库的分类

分 类	说 明
输出（Outputs）	输出指定的信息或绑定值，其名称 output 作为开头
输入（Inputs）	提供进行输入，其名称以 input 作为开头
命令（Commands）	提供命令或连接按钮，其名称以 command 作为开头
选择（Selections）	提供选项的选取，其名称以 select 作为开头
其他	包括了 form、message、messages、graphicImage 等未分类的标签

JavaServer Faces 标准 HTML 标签包括了几个共同的属性，如表 14-2 所示。

表 14-2 JSF 的 UI 组件的共同属性

属 性	适 用	说 明
id	所有组件	组件的唯一标识
binding	所有组件	绑定其他 UI 组件
rendered	所有组件	是否显示组件
式样表（styleClass）	所有组件	指定组件的层叠式样表
style	所有组件	用分号分隔的层叠样式表声明列表。这些声明以特定 HTML 标记的样式属性传递到 HTML 代码。可以使用样式编辑器来定义该属性的值
text	输入、输出、命令组件	组件的值
valueChangeListener	输入组件	用于设置组件的值发生变化时的处理程序
converter	输入、输出组件	设定转换器
validator	输入组件	设定验证器
required	输入组件	是否验证必填栏位
immediate	输入、命令组件	是否为立即事件

14.1.2 安装 Visual Web 开发包

在 NetBeans 5.x 中，进行可视化的 JSF 开发，需要单独下载 NetBeans Visual Web Pack 开发包，并进行安装。在 NetBeans 6.0 中，图 14-1 所示的两个安装文件中都包含了可视化的 JSF 开发功能。

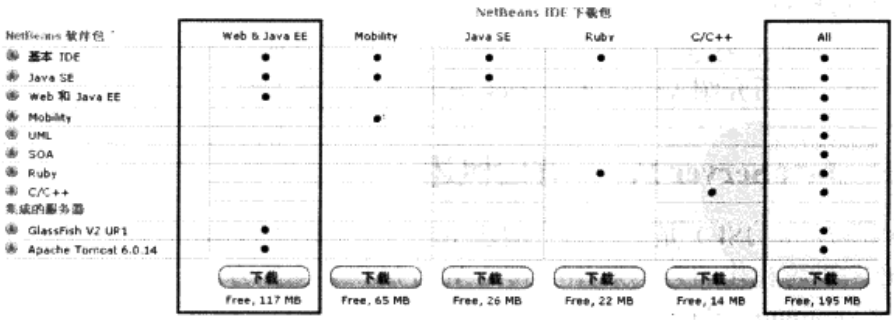


图 14-1 下载包含可视化的 JSF 开发功能的安装程序

14.2 创建简单的 Visual Web 应用程序

安装了 NetBeans Visual Web Pack 开发包后，就可以进行可视化的 Web 应用程序开发了。本节将开发一个简单的可视化 Web 应用程序。

Download at http://www.pin5i.com/

14.2.1 创建可视化的 Web 应用程序项目

在 NetBeans 集成开发环境中，选择“文件”→“新建项目”命令，打开“新建项目”对话框。在“类别”列表中选择“Web”节点，在“文件类型”列表中选择“Web 应用程序”节点。单击“下一步”按钮，打开“新建 Web 应用程序”对话框。该对话框的设置方法已经在第 11 章中介绍过，这里不再赘述。设置完毕后，单击“下一步”按钮，打开“框架”对话框，如图 14-2 所示。

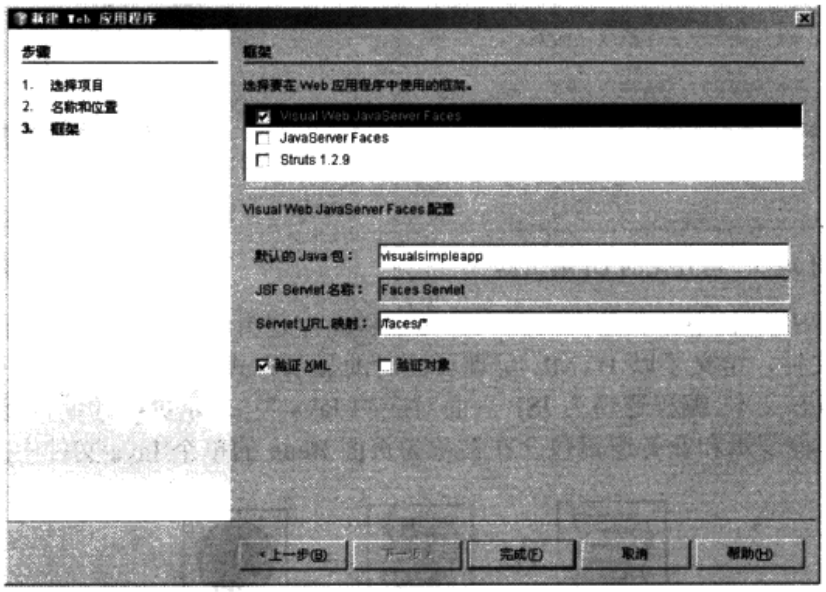


图 14-2 “框架”对话框

选中“Visual Web JavaServer Faces”复选框。在下面的“默认的 Java 包”文本框中输入 Java 包的名字。单击“完成”按钮，即可创建一个支持可视化 JSF 组件开发的 Web 应用程序项目。

14.2.2 可视化的 Web 应用程序开发环境

创建可视化的 Web 应用程序项目后，在“项目”视图中会显示可视化 Web 项目的结构，如图 14-3 所示。展开项目的“Web 页”节点，会显示创建的 JSP 文件。双击自动创建的 Page1.jsp 文件，会在“导航”视图中显示该页面的内容，如图 14-4 所示。

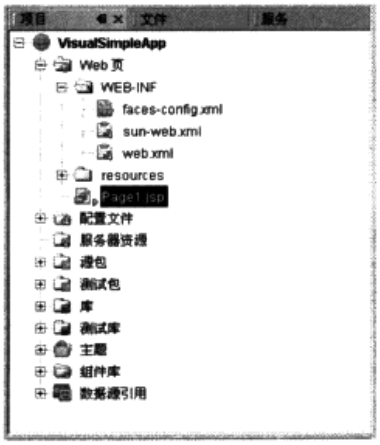


图 14-3 可视化 Web 项目的“项目”视图

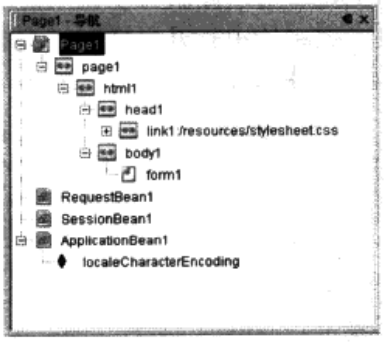


图 14-4 可视化 Web 项目的“导航”视图

Download at http://www.pin5i.com/

打开 Page1.jsp 文件时，会在源代码编辑器中打开可视化的 UI 编辑器，如图 14-5 所示。此时，可以从如图 14-6 所示的组件面板中拖曳 UI 组件到页面中。在源代码编辑器中，单击“JSP”标签可以修改页面对应的 JSP 代码；单击“Java”标签可以修改页面对应的页面 Bean 模型。

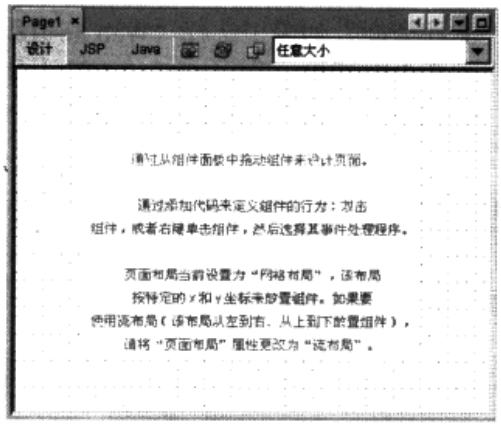


图 14-5 可视化的 UI 编辑器

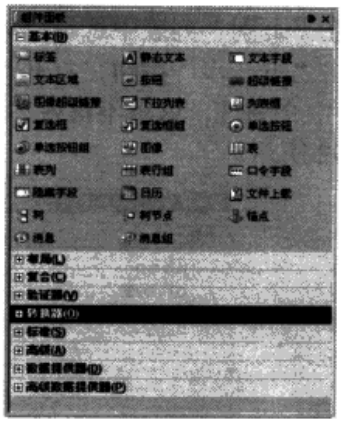


图 14-6 组件面板

在可视化 Web 项目中，页面由页面表示元素和编程逻辑组成，如图 14-7 所示。其中，表示元素为 JSP 文件，定义了以 HTML 呈现页面时的显示方式。页面是向用户显示的界面，可以包含组件和静态文本。编程逻辑为 JSP 页面对应的 Java 类，表示页面的编程逻辑和业务逻辑。默认情况下，编程逻辑和业务逻辑包含在被称为页面 Bean 的单个 Java 文件中。



图 14-7 页面组成



说明：NetBeans 集成开发环境自动将 JSP 页与页面 Bean 中的 Java 代码保持同步。例如，如果在 JSP 页面中添加一个组件，将会自动在页面 Bean 文件中进行反映；相反，如果页面 Bean 的 Java 文件中进行了某种更改，JSP 页面文件中也会反映此更改。



14.2.3 进行可视化的页面编辑

在可视化的 Web 项目中，打开页面文件时，会打开组件面板。组件面板通常会位于集成开发环境的右上角，用于提供可以在当前选定页面或代码编辑器中使用的组件和代码片段。当在可视化的页面编辑器中打开页面时，会显示如图 14-6 所示的界面，提供了可以在页面中使用的组件。这些组件的分类以及说明如表 14-3 所示。

表 14-3 组件分类及说明

分 类	说 明
基本	包含最常用的组件，例如按钮、标签、文本字段、复选框、图像、表、下拉列表以及文本字段等
布局	包含组织页面的组件，例如在网格中排列组件的网格面板组件，或者将组件划分为组的组面板组件
组合	具有更复杂的组织结构和专门用途的组件，如日历组件（用于选择日期）、警报组件（用于显示带格式的警告信息，通常位于页面的顶部）

Download at http://www.pin5i.com/

续表

分　类	说　明
验证器	一组可用来验证用户输入的数据的数据检验器
转换器	一组可以在组件中使用的数据转换器，可以根据需要，通过它们在 String 数据类型和其他数据类型之间进行转换
标准组件	使用 HTML 标记的一组原始的标准组件
高级	为高级用户提供的一组 JavaServerPages 标记，例如编码组件（设置响应字符编码）；脚本组件（是 HTML Script 标记的实现）
数据提供器	提供数据源接口的一组组件（如数据库表和 JavaBeans 对象）
高级数据提供器	一组仅供高级用户使用的数据提供器

当在源代码编辑器中编辑页面对应的 Java 文件时，组件面板中会显示可以使用的代码片段，如图 14-8 所示。可以将这些代码片段添加到页面 Bean、JSP 页以及与页面关联的其他代码文件中。

在类别上单击鼠标右键，选择弹出菜单中的“创建新的代码片段”命令，可以创建一个新代码片段。右键单击某个代码片段，选择弹出菜单中的“编辑代码片段”命令，会打开如图 14-9 所示的对话框，进行代码片段的编辑和修改。



图 14-8　组件面板中的代码片段

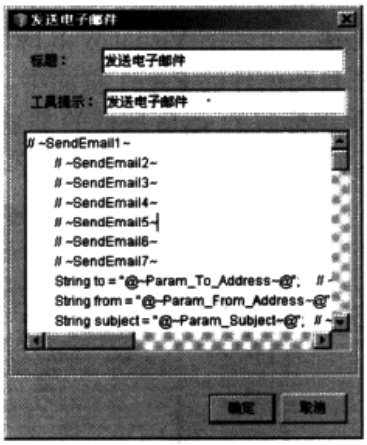


图 14-9　编辑代码片段

这里，在可视化的页面编辑器中打开 Page1.jsp 页面，然后从组件面板向页面中添加一个“标签”组件，如图 14-10 所示。单击标签组件，在变为编辑状态后，输入“你好，世界!”，如图 14-11 所示。

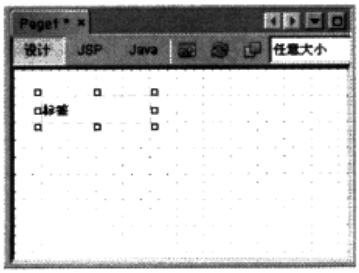


图 14-10　添加一个“标签”组件

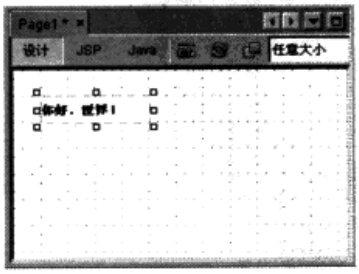


图 14-11　编辑标签的显示值

Download at http://www.pin5i.com/

在保存修改后，就可以运行项目了。项目的运行界面如图 14-12 所示。至此，创建了一个简单的可视化的 Web 应用程序。



图 14-12　程序的运行界面

14.3　进行页面导航

在 Web 应用程序中，可能经常需要在页面之间进行切换，即在一个页面中单击某个链接或者按钮，会打开另外一个页面。本节将介绍几种在页面之间进行导航的方法。

14.3.1　简单的页面导航

本节将建立一个简单的可视化的 Web 应用程序，在程序中创建三个 Web 页面，然后设置三个页面之间的导航。单击页面上的按钮，会切换到下一个页面。

❶ 创建一个新的 Visual Web 应用程序项目，命名为 PageNavigation，并设置为主项目。

❷ 在自动创建的 Page1 页面中，添加一个标签，设置其 text 属性为“第 1 页”。在“属性”视图中，单击标签的 style 属性的 按钮，打开 style 编辑器，如图 14-13 所示。选中“使用值”单选按钮，在左边的列表中选择“字体”，然后设置字体大小为 18。

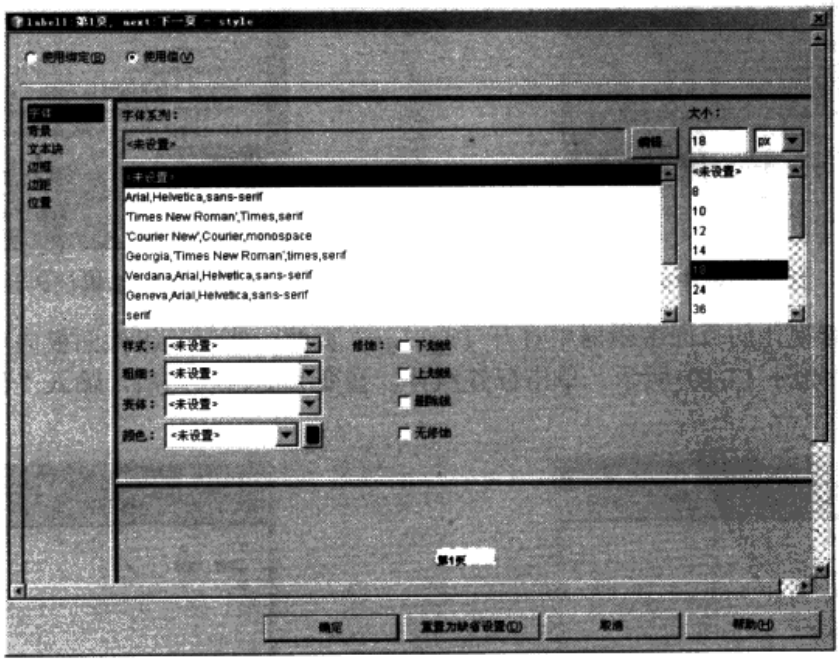


图 14-13　style 编辑器

❸ 在自动创建的 Page1 页面中，添加一个命令按钮。设置其 id 属性为 next，其 text 属性为“下一页”。Page1 页面的显示如图 14-14 所示。

Download at http://www.pin5i.com/

❹ 新建页面 Page2 和 Page3，按照步骤❷和步骤❸的方法添加界面组件，如图 14-15 和图 14-16 所示。

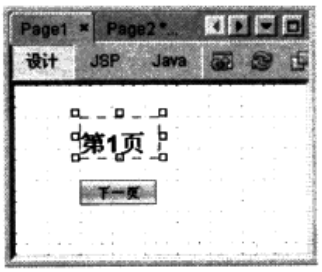


图 14-14 Page1 设计界面

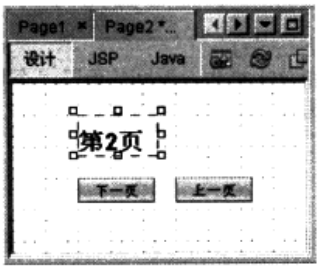


图 14-15 Page2 设计界面

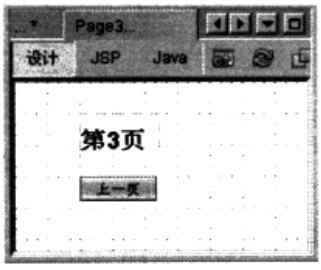


图 14-16 Page3 设计界面

❺ 在页面的空白处单击鼠标右键，选择弹出菜单中的“页面导航”命令，或者在“项目”视图的项目节点下双击“页面导航”节点，打开可视化的页面导航设计器，如图 14-17 所示。

❻ 单击 Page1.jsp 图标，选择其中按钮旁边的小方框，将其拖曳到 Page2.jsp，如图 14-18 所示。释放鼠标后，就建立了一个从 Page1.jsp 到 Page2.jsp 的导航线路。

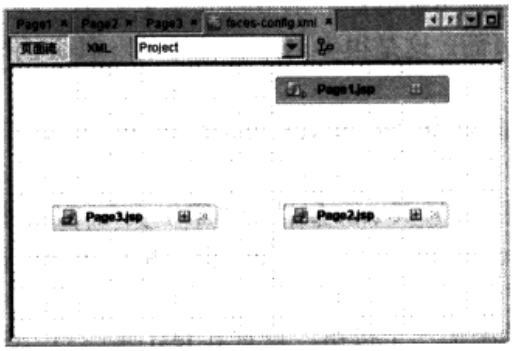


图 14-17 可视化的页面导航设计器

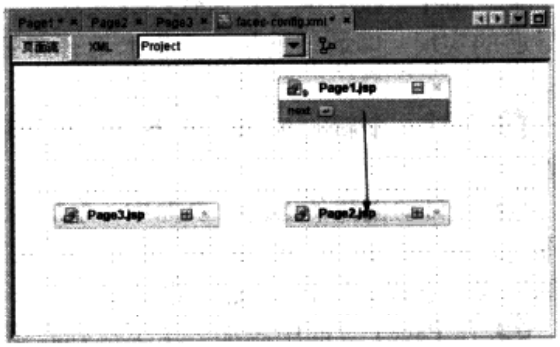


图 14-18 建立导航线路

❼ 在 Page2.jsp 中有两个按钮，因此分别建立指向 Page1.jsp 和 Page3.jsp 的导航线路。在 Page3.jsp 中，把导航线路指向 Page2.jsp。设计好的页面间的导航路线如图 14-19 所示。页面间的导航线路也是有名字的，例如图 14-19 中，页面 Page1.jsp 到 Page2.jsp 间的线路名字为 case1（对于页面 page1.jsp 而言）；Page2.jsp 到 Page3.jsp 的导航线路名字为 case2（对于页面 page2.jsp 而言）。单击某个导航线路，可以在属性窗口中修改线路的名字。

❽ 至此，完成了一个页面导航的简单例子。运行程序后，单击页面上的按钮就可以在页面间进行切换了。

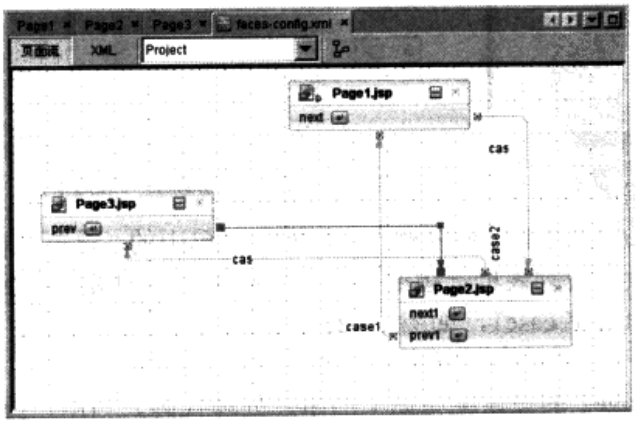


图 14-19 完整的页面间导航路线

Download at http://www.pin5i.com/

注意：通过页面导航设计器进行导航设计后，不需要再对按钮的 action 事件进行编程了。此时如果双击按钮，会自动添加如下的代码：

```
public String next_action() {
    // 待做事项：处理按钮单击操作。返回的值是一个导航
    // 条件名称，如果它为 Null，则返回到同一页

    return "case1";
}
```

其中，函数的返回值就是希望执行的导航线路名字。如果返回值为 null，则返回到当前页面中。此时，如果在把按钮的 action 事件处理代码除掉，那么运行时再单击该按钮，将不再执行导航。当遇到这种问题时，可以把不工作的导航线路删除，然后重新建立，即可重新进行导航了。

14.3.2 通过下拉列表框进行页面导航

上一节中，没有编写任何代码就实现了一个简单的页面导航。本节将介绍更复杂一些的页面导航的实现，即通过下拉列表框来选择要打开的页面。这是很多网站都使用的一种页面导航方式。

❶ 创建一个新的 Visual Web 应用程序项目，命名为 PageNavigation_Dropdown_Button，并设置为主项目。

❷ 在自动创建的 Page1 页面中，添加一个标签，设置其 text 属性为“第 1 页”。在“属性”视图中，单击标签的 style 属性的 按钮，打开 style 编辑器。选中“使用值”单选按钮，在左边的列表中选择“字体”，然后设置字体大小为 18。

❸ 在 Page1 页面中添加一个下拉列表组件。再添加一个按钮，设置其 text 属性为“Go”，如图 14-20 所示。

❹ 新建页面 Page2，按照步骤❷和步骤❸的方法添加界面组件，如图 14-21 所示。

❺ 在页面的空白处单击鼠标右键，选择弹出菜单中的“页面导航”命令，或者在“项目”视图的项目节点下双击“页面导航”节点，打开可视化的页面导航设计器。两个页面之间的导航设置如图 14-22 所示。

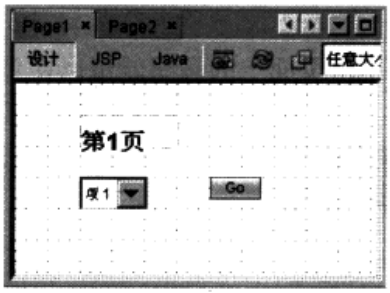


图 14-20 Page1 的界面设置

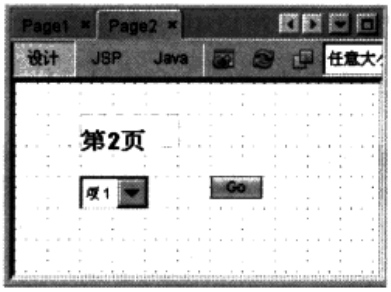


图 14-21 Page2 的界面设置

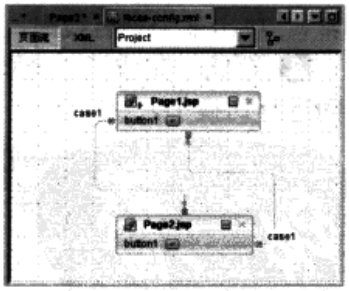


图 14-22 两个页面之间的导航设置

❻ 右键单击 Page1 中的下拉列表框，选择弹出菜单中的“设置默认选项”命令，打开“选项定制器”对话框修改下拉列表框选项的值。当选择页面 2 时要导航到第 2 页，因此把选项 2 的值设置为导航线路的名字 case1，如图 14-23 所示。

❼ 设置 Page2 中下拉列表框的选项值，如图 14-24 所示。

❽ 为 Page1 和 Page2 的按钮编写如下的事件处理代码。当用户改变下拉列表框的值后，单击按钮会打开指定的页面。

Download at http://www.pin5i.com/

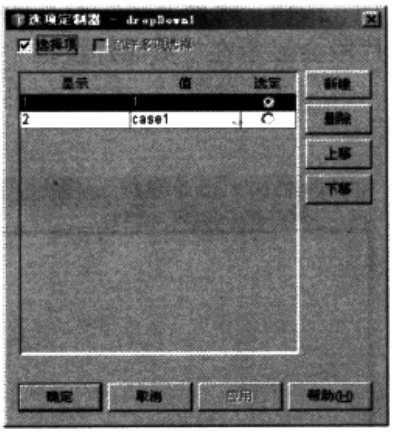


图 14-23 Page1 中的下拉列表框的选项值

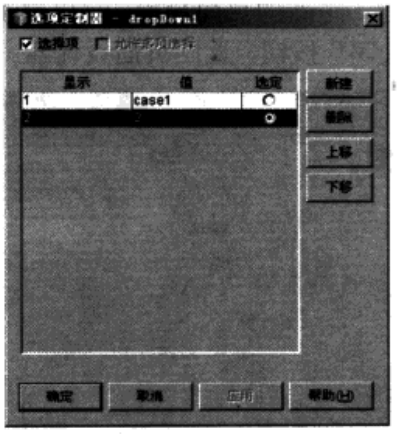


图 14-24 Page2 中的下拉列表框的选项值

```
public String button1_action() {
    // 待做事项：处理按钮单击操作。返回的值是一个导航
    // 条件名称，如果它为 Null，则返回到同一页

    return (String) dropDown1.getValue();
}
```

❾ 至此，完成了一个比较复杂的页面导航。运行程序后，单击通过下拉列表框选择页码，然后单击页面上的按钮就可以在页面间进行切换了。

14.4 使用页面片段

在 JSF 组件中，有一个页面片段框组件，可以用来使用代码片段。JSF 的页面片段，就好像程序开发中的函数一样，可以把一些经常使用的功能封装在一起，以便重复使用。例如在一个公司网站中，要在每个页面的顶部显示公司的图标，因此可以把这个代码封装为一个页面片段，然后在每个页面中静态引用这个页面片段。当然，这个只显示图像的页面片段有些简单，实际项目中可以实现更复杂一些的功能封装。下面介绍这个公司图标页面片段的实现方法，以及在页面中的引用方法。

❶ 创建一个新的 Visual Web 应用程序项目，命名为 FragmentDemo，并设置为主项目。

❷ 选择“新建”→“新建文件”命令，打开“新建文件”对话框。在“类别”列表中选择“Visual Web”节点，然后在“文件类型”列表中选择“Visual Web JSF 页面片段”节点。单击“下一步”按钮，打开“新建页面片段”对话框，如图 14-25 所示。

❸ 在“文件名”文本框中输入页面片段的文件名，这里输入 FragmentTitle，文件以 jspf 作为扩展名。单击“完成”按钮，创建一个空白的页面片段。在页面编辑器中可以对此页面片段进行修改，如图 14-26 所示。其中，白色的区域为页面片段的实际大小。

❹ 从组件面板中向白色区域添加一个图像组件。右键单击该组件，选择弹出菜单中的“设置图像”命令，打开“图像定制器”对话框。在该对话框中选择要显示的图像，如图 14-27 所示。

单击“确定”按钮，此时图像组件中会显示选中的图像，如图 14-28 所示。

❺ 选中页面片段，在属性窗口中设置其高度和宽度，使得大小正好符合图像组件的大小。至此，完成了一个页面片段的设置。下面将在页面中使用这个页面片段。

❻ 在自动创建的 Page1 页面中，从组件面板的“布局”分类中，添加一个“页面片段框”组件。右键单击该组件，选择弹出菜单中的“选择页面片段”命令，打开“选择页面片段”对话框。前面创建的页面片段会显示在“页面片段”下拉列表框中，如图 14-29 所示。此时，如

Download at http://www.pin5i.com/

果要创建一个新的页面片段，可以单击“创建新的页面片段”按钮。选择要显示的页面片段后，单击“关闭”按钮，返回页面设计器。此时，选中的页面片段会显示在“页面片段框”组件中。

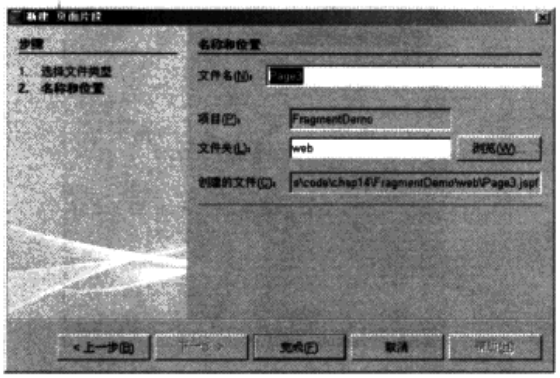


图 14-25 “新建页面片段”对话框

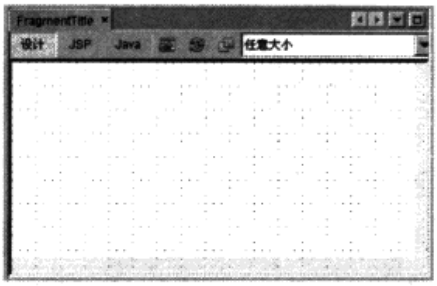


图 14-26 创建一个空白的页面片段

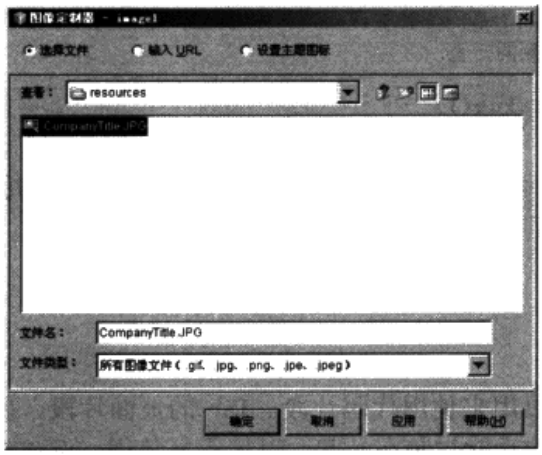


图 14-27 “图像定制器”对话框



图 14-28 添加了图像组件的页面片段



图 14-29 “选择页面片段”对话框

使用同样的方法，可以在不同的页面中使用已经创建的页面片段，从而达到代码的重复利用，提高开发效率。

14.5 使用 JSF 实现登录模块

前面的几节介绍了 JSF 组件的开发方法，以及几个简单的实例。下面将介绍如何利用 JSF 技术实现登录模块。在这个登录模块中，登录界面、登录成功界面、登录失败界面都使用 JSF 组件完成，身份验证通过 Java Bean 进行判断。创建一个 JSF_Login 的 Visual Web 应用程序，并设置为主项目，下面的操作都在这个项目中进行。

14.5.1 实现登录失败的界面

新建一个页面，命名为 Error。从 JSF 组件面板中，向该页面添加一个“标签”组件。双击该组件，输入“登录失败”。在属性窗口中设置，双击 Style 属性，把字体大小设置为 24。

从 JSF 组件面板中，向该页面添加一个“图像”组件。右键单击该组件，选择弹出菜单中的“设置图像”命令。在打开的“图像定制器”对话框中，从以前的登录模块项目中选择 error.jpg 文件。

从 JSF 组件面板中，向该页面添加一个“超级链接”组件。双击该组件，输入“返回”。

Download at http://www.pin5i.com/

在页面的空白处单击鼠标右键，选择弹出菜单中的“页面导航”命令，打开可视化的页面导航编辑器。单击 Error.jsp 页面，从 hyperlink1 拖曳一个连接到 Page1.jsp，如图 14-30 所示。这样当单击该超级链接后，会自动返回到登录界面。

创建完成的 Error 页面显示如图 14-31 所示。

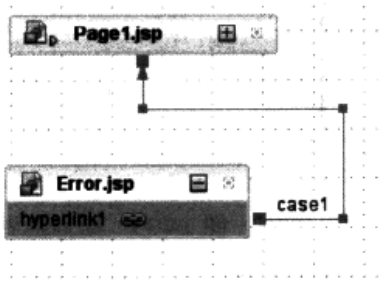


图 14-30 添加通过超级链接组件实现的从 Error.jsp 页面到 Page1.jsp 的导航



图 14-31 Error 页面

14.5.2 实现登录成功的界面

新建一个页面，命名为 Welcome。从 JSF 组件面板中，向该页面添加一个“标签”组件。双击该组件，输入“登录成功”。在属性窗口中设置，双击 Style 属性，把字体大小设置为 24。

从 JSF 组件面板中，向该页面添加一个“标签”组件。在属性窗口中，设置其 id 属性为 lblMsg。该组件用于在登录成功后，显示登录用户的名字。打开 SessionBean1.java 文件，添加一个 String 类型的字段，名字为 username。添加的代码如下所示：

```
private String username;
…
    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }
```

打开 Welcome 类，在 prerender 方法中添加如下的代码：

```
    public void prerender() {
        SessionBean1 sb = getSessionBean1();
        String un = sb.getUsername();
        lblMsg.setText("欢迎您：" + un);
    }
```

其中，getSessionBean1()用于返回当前的 SessionBean1 对象。getUsername 方法返回当前登录的用户名，该值在用户登录成功后在登录界面 Page1 中被添加。与这段代码对应的代码，将在 14.5.3 中进行介绍。

从 JSF 组件面板中，向该页面添加一个“图像”组件。右键单击该组件，选择弹出菜单中的“设置图像”命令。在打开的“图像定制器”对话框中，从以前的登录模块项目中选择 welcome.jpg 文件。创建完成的 Welcome 页面显示如图 14-32 所示。

Download at http://www.pin5i.com/

14.5.3 登录界面以及身份验证

在自动创建的 Page1 页面中，添加一个“文本字段”组件和一个“口令字段”组件，再添加两个“按钮”组件。组件的界面布局如图 14-33 所示，组件的属性如表 14-4 所示。



图 14-32 Welcome 页面

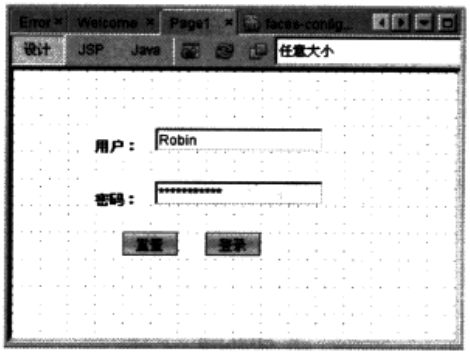


图 14-33 组件的界面布局

表 14-4 组件属性设置

组 件	属 性	属性值
文本字段	id	fldUsername
	label	用户：
	text	Robin
口令字段	id	fldPassword
	label	密码：
按钮	id	reset
	text	重置
	reset	true
按钮	id	submit
	text	登录

在页面的空白处单击鼠标右键，选择弹出菜单中的“页面导航”命令，打开可视化的页面导航编辑器。设置页面之间的导航线路，如图 14-34 所示。选中由 Page1 到 Error 页面的导航线路，在属性编辑器中，将该线路的名称修改为 Error。选中由 Page1 到 Welcome 页面的导航线路，在属性编辑器中，将线路的名称设置为 Welcome。

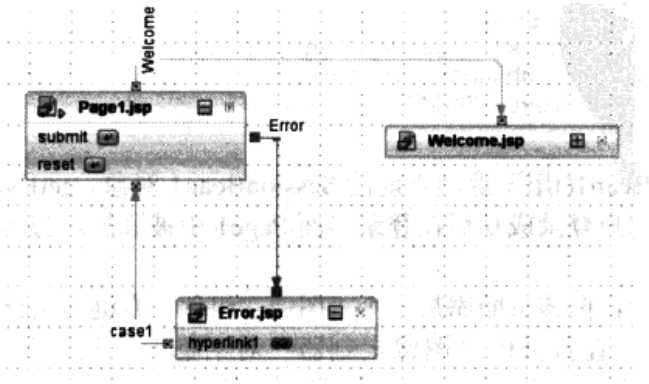


图 14-34 设置页面之间的导航线路

Download at http://www.pin5i.com/

把以前的登录模块中用于身份验证的 LoginBean 文件，复制到当前项目的源代码目录中。在本实例中，目录为 JSF_Login\src\java\jsf_login。把 LoginBean 包的名字修改为当前项目中使用的名字，这里修改为 jsf_login。双击“登录”按钮，添加如下的代码：

```
public String submit_action() {
    // 待做事项：处理按钮单击操作。返回的值是一个导航
    // 条件名称，如果它为 Null，则返回到同一页
    String username = (String)fldUsername.getText();
    String password = (String)fldPassword.getText();

    //以下代码利用 LoginBean 进行登录判断
    boolean loginflag = false;
    LoginBean bean = new LoginBean();
    bean.setUsername(username);
    bean.setPassword(password);
    loginflag = bean.checklogin();

    //根据登录情况打开不同的页面（视图）
    if (loginflag) {
        SessionBean1 sb = getSessionBean1();
        sb.setUsername(username);
        return "Welcome";
    }
    return "Error";
}
```

代码首先返回用户输入的用户名和密码，然后通过 LoginBean 进行身份验证。如果验证成功，则通过 sb.setUsername 语句保存用户名，这样就可以在欢迎界面中取出该值进行显示了。

如果登录成功，则返回导航 Welcome，打开欢迎界面；否则返回导航 Error，打开登录错误页面。至此，完成了一个通过 JSF 组件实现的简单的登录模块。

14.6 进行 AJAX 开发

AJAX 是这几年比较热门的一种 Web 客户端技术，很多知名的网站都使用了该技术。例如国外著名的搜索引擎网站 Google、电子交易网站 EBAY 等；国内如 163、新浪等。NetBeans 提供了对 AJAX 开发的良好支持，可以进行可视化的开发。可以说 NetBeans 是目前支持 AJAX 最好的集成开发环境。

14.6.1 下载并导入 AJAX 及其组件

AJAX 组件并没有与集成开发环境一起发布，需要从更新中心下载并导入。NetBeans 5.x 和 NetBeans 6.0 的安装过程有所不同。

在 NetBeans 5.x 中选择“工具”→“更新中心”命令，打开“更新中心向导”对话框。单击“下一步”按钮，连接到更新中心查找可以利用的更新新模块。查找完毕后，显示如图 14-35 所示的对话框。

在“NetBeans Visual Web Pack 更新中心”节点下，可以看到“BluePrints AJAX Components”节点。选中该节点，单击“添加”按钮，将其移动至“包含在安装中”列表中。单击“下一步”按钮按照提示完成安装。

安装完毕后，需要重新启动 NetBeans 集成开发环境。在安装了 AJAX 模块后，需要将其导入到组件库中，才能够被使用。在 NetBeans 集成开发环境中，选择“工具”→“组件库管理器”命令，打开“组件库管理器”对话框。单击“导入”按钮，打开“导入组件库”对话框。

Download at http://www.pin5i.com/

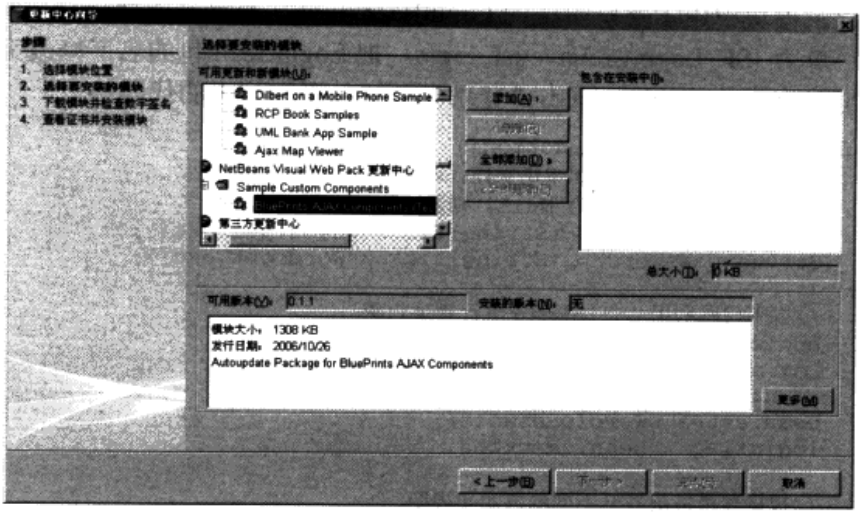


图 14-35 “选择要安装的模块”对话框

通过“导入组件库”对话框，可以将新的 JavaServer Faces 组件添加到 NetBeans 集成开发环境中。单击“浏览”按钮，查找包含要导入的组件的有效.complib 文件。这里选择 ui.complib 文件，然后单击“打开”命令。ui.complib 路径和 BluePrint AJAX 组件将出现在“导入组件库”对话框中，如图 14-36 所示。

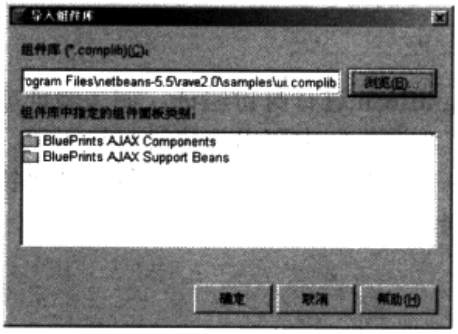


图 14-36 “导入组件库”对话框中

单击“确定”按钮，完成导入。如果以前已经安装了 BluePrints AJAX 组件，将会显示一个对话框，询问是否要覆盖组件库。如果看到此对话框，请单击“是”按钮。导入完毕的“组件库管理器”对话框，如图 14-37 所示。单击“关闭”按钮，完成 AJAX 组件的导入，返回 NetBeans 集成开发环境。

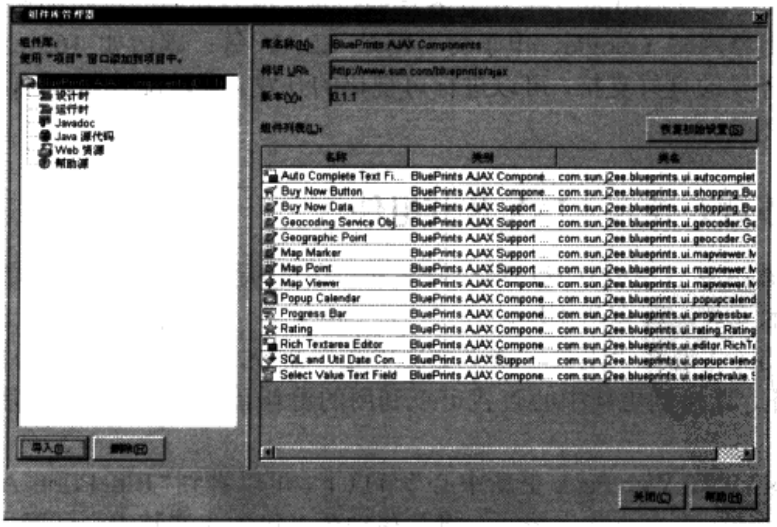


图 14-37 “组件库管理器”对话框

NetBeans 6.0 中 AJAX 组件的安装有所简化。在如图 14-38 所示的“插件”对话框中，选择“jMaki Ajax support”模块，单击“安装”按钮即可完成模块的安装。安装完毕后重新启动 IDE 即可使用 AJAX 组件了。

Download at http://www.pin5i.com/

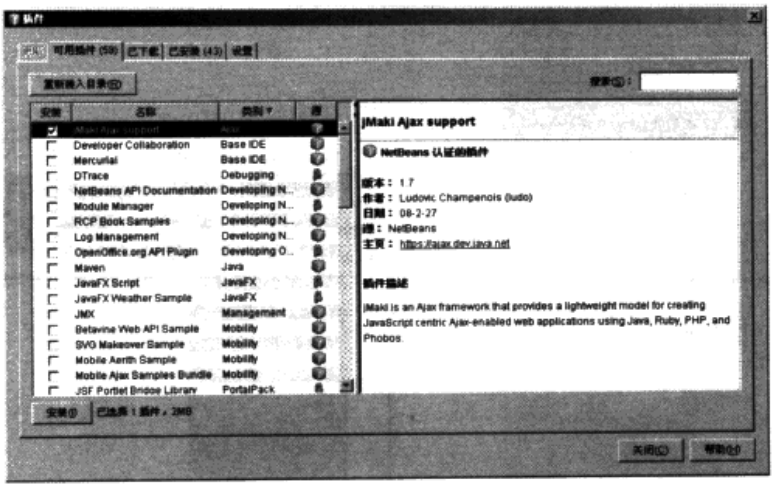


图 14-38 “插件”对话框

14.6.2 AJAX 开发实例：在页面中使用日历

在安装 AJAX 模块后，就可以在项目中使用 AJAX 组件了。下面创建一个名字为 AjaxDemo 的 Web 项目，并设置为主项目。在如图 14-39 所示的“框架”对话框中，要选择使用“jMaki Ajax Framework”。在 CSS Layout 中选择“Standard”布局。

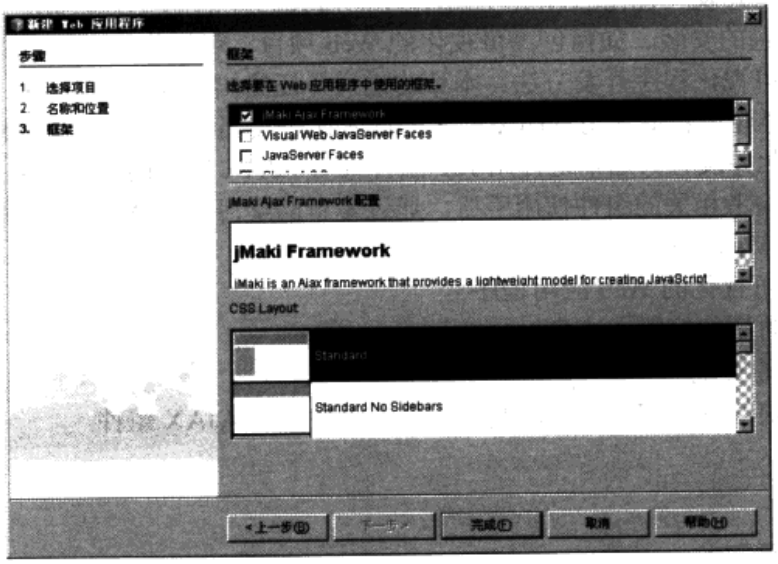


图 14-39 “框架”对话框

创建项目后，打开自动创建的 index.jsp 文件。此时的组件面板如图 14-40 所示。

从组件面板的 jMaki Dojo 中拖曳“Date Picker”组件和“Drop Down Date Picker”组件，将自动添加如下所示的两行代码：

```
<a:widget name="dojo.datepicker" />
<a:widget name="dojo.dropdowndatepicker" />
```

运行程序显示界面如图 14-41 所示。

至此，完成了一个在 Web 项目中使用 AJAX 组件的简单示例。至于其他 AJAX 组件的使用方法，不在本书的介绍范围之内，请读者自行参考相关文档。

Download at http://www.pin5i.com/

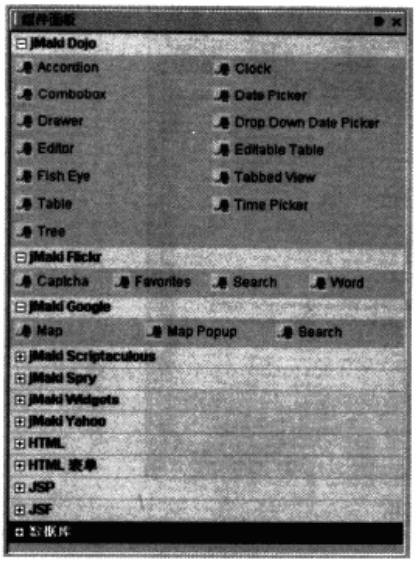


图 14-40 AJAX 组件面板

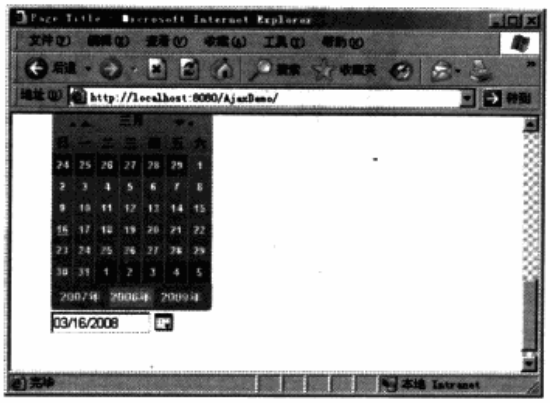


图 14-41 程序显示界面

14.7 小结

本章主要如何开发可视化的 Web 应用程序，内容的介绍主要围绕 JSF 组件展开。包括：Visual Web 开发包的安装、如何创建可视化的 Web 项目、可视化的 Web 开发环境，以及几个简单并且常用的可视化程序开发方法。本章的最后，还给出了在可视化项目中如何进行 AJAX 开发，这部分内容对于希望使用 AJAX 技术的读者比较实用。由于本书的重点是介绍 NetBeans 的使用方法，因此并没有大篇幅地介绍 JSF 组件或者 AJAX 组件的具体使用方法，所以感兴趣的读者可以自己参考相关的组件使用手册。通过本章的学习，读者应该掌握以下基本内容：

- 如何安装 Visual Web 开发包
- 如何创建可视化的 Web 应用程序
- JSF 组件的简单使用方法
- 页面导航的实现
- AJAX 模块的安装和导入，以及如何在项目中使用 AJAX 组件

Download at http://www.pin5i.com/

第 15 章

企业级应用程序开发

在前面的章节中介绍了使用 NetBeans 开发桌面应用程序、Web 应用程序、可视化的 Web 应用程序等。从本章开始将介绍企业级应用程序的开发，及其相关技术。本章将首先介绍一个简单的企业级应用程序的开发方法，以便于读者掌握程序开发的大致步骤。然后在此基础上进一步扩展，介绍 EJB 模块和持久性 API 的开发方法。最后，简单给出了几种 Sun 资源的创建方法。

本章主要内容

- 如何创建一个简单的企业级应用程序
- 会话 Bean 有哪两种类型，以及各自的应用场合
- 会话 Bean、实体 Bean 和消息驱动 Bean 的区别
- 如何通过数据库创建实体 Bean
- 如何创建实体类的会话 Bean
- 如何创建消息 Bean，以及如何处理消息
- 如何创建持久性单元

本章亮点

- 如何在客户端发送 JMS 消息
- 如何通过实体类创建 JSF 页
- 在 Web 页面中调用 EJB 模块

15.1 创建一个简单的企业级应用程序

企业应用程序是由 EJB 模块和 Web 应用程序模块组成的集合，这些模块在部署到 Java EE 应用服务器后会配合使用，而企业应用程序中包含了有关模块之间如何配合使用的信息。另外，企业应用程序还包含了有关模块与部署了企业应用程序的应用服务器之间如何配合使用的信息。

15.1.1 创建企业应用程序项目

企业应用程序没有自身的源文件，它只包含部署描述符和其他配置文件。在编译时，将生成每个企业应用程序模块的归档文件（JAR 文件和 WAR 文件），并将其组装到一个企业归档（Enterprise Archive, EAR）文件中。此文件将被部署到应用服务器中。

在进行企业级程序开发前，请首先参考第 6 章的内容，配置好 Java EE 应用程序服务器。下面以一个简单的演示程序说明开发企业级应用程序的一般方法。

在 NetBeans 集成开发环境中，选择“文件”→“新建项目”命令，打开“新建项目”对话框。在“类别”列表中选择“企业”节点，在右边的“项目”列表中选择“企业应用程序”节点。单击“下一步”按钮，打开“新建 企业应用程序”对话框，如图 15-1 所示。

Download at http://www.pin5i.com/

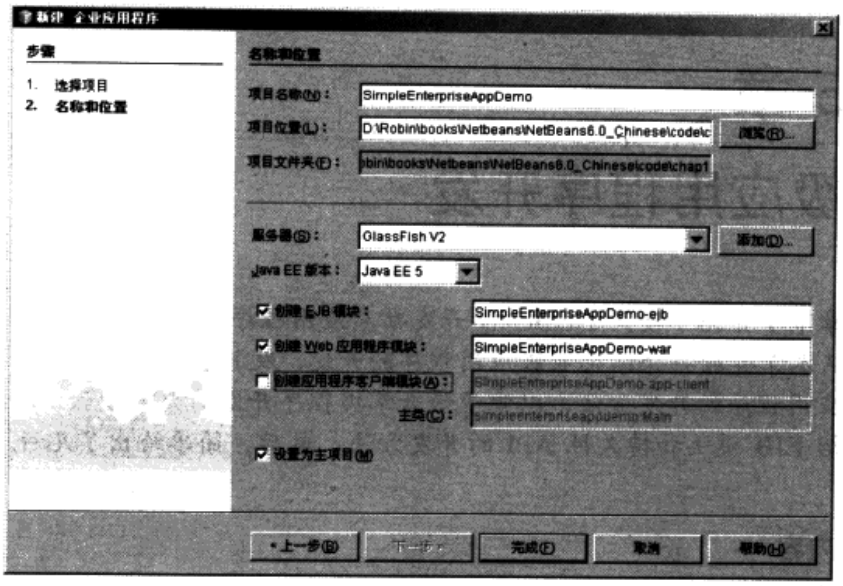


图 15-1 “新建 企业应用程序”对话框

在“项目名称”文本框中输入项目的新名字 SimpleEnterpriseAppDemo，在“项目位置”文本框中输入项目所在的文件夹位置。

在“服务器”下拉列表框中选中 Glass Fish 服务器。在“Java EE 版本”下拉列表框中选择“Java EE 5”。

选中“创建 EJB 模块”复选框，在后面的文本框中输入 EJB 模块的名字；选中“创建 Web 应用程序模块”复选框，在后面的文本框中输入 Web 应用程序模块的名字。

“创建应用程序客户端模块”复选框用于指定是否创建应用程序客户端。Java EE 应用程序客户端是独立的应用程序，它被配置作为 Java EE 企业应用程序的一部分进行工作。当然，任何 Java 应用程序都可以访问远程 EJB 方法和 Web 服务，因此，不是一定需要创建 Java EE 应用程序客户端来访问企业应用程序。通过企业应用程序客户端访问企业应用程序的优点是，应用程序客户端可以访问在企业应用服务器上运行的服务和功能（例如，可以为企业应用程序客户端配置安全角色和权限），而常规 Java 应用程序则不能。这里因为只创建一个简单的企业级应用程序，所以取消选择“创建应用程序客户端模块”复选框。

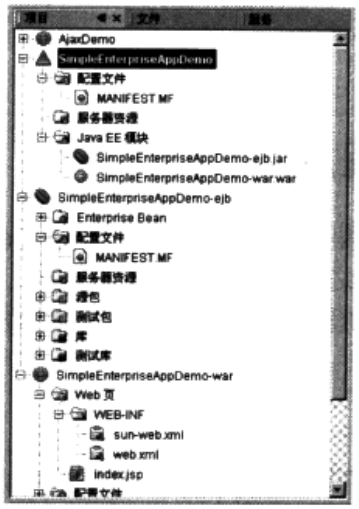


图 15-2 “项目”窗口中的企业应用程序

单击“完成”按钮，将创建企业应用程序项目、Web 项目和 EJB 模块项目，如图 15-2 所示。其中，企业应用程序项目和 Web 项目包含了 Java EE 5 部署描述符。EJB 模块项目没有任何部署描述符，因为在 EJB 3.0 中不需要描述符，所有的元数据都写在声明中。



15.1.2 实现 EJB 3.0 Bean

在创建企业应用程序项目后，要实现 EJB 模块。在 EJB 模块中会封装各种的业务逻辑，在部署 EJB 模块后，就可以在客户端调用 EJB 完成必要的功能。

Download at http://www.pin5i.com/

在“项目”窗口中，右键单击 EJB 模块项目节点，这里右键单击“SimpleEnterpriseAppDemo-ejb”节点。选择弹出菜单中的“新建”→“会话 Bean”命令，打开“新建 会话 Bean”对话框，如图 15-3 所示。

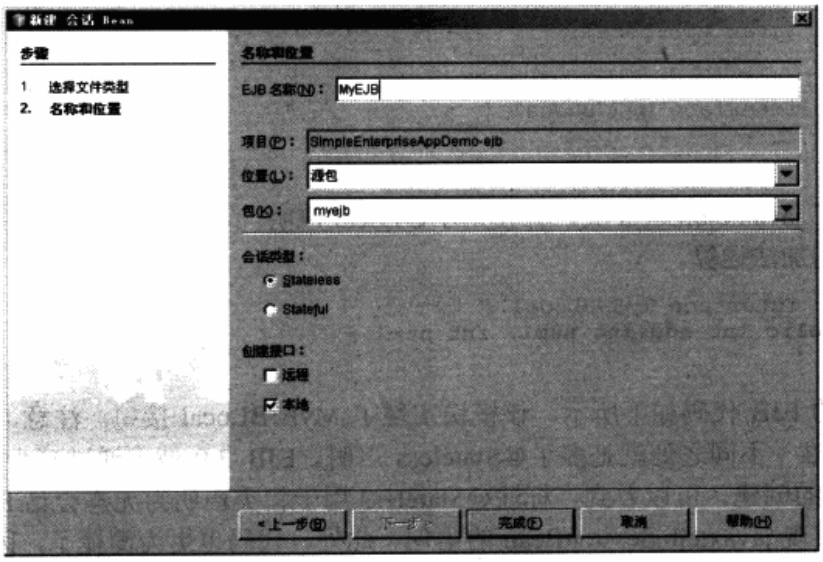


图 15-3 “新建 会话 Bean”对话框

在“EJB 名称”文本框中输入 EJB 的名字 MyEJB，该名字将添加到部署描述符中，并作为会话 Bean 类和接口名称的基础。在“位置”下拉列表框中选择要创建的会话 Bean 文件的源文件夹。在“包”下拉列表框中输入新的包名字 myejb。

会话 Bean 有以下两种类型。

- 无状态会话 Bean：选中 Stateless 单选按钮，会创建一个无状态会话 Bean。该 Bean 在与用户的会话期间不保存状态信息。对于在客户端和应用程序服务之间的简单交互，不需要保存状态时，可以使用这种类型的 Bean。例如，进行某种计算的 Enterprise Bean 就是无状态的会话 Bean，调用该 Bean 时只要给出必要的参数，就可以返回计算的结果。
- 有状态会话 Bean：选中 Stateful 单选按钮，会创建一个有状态会话 Bean。该 Bean 在与用户的会话期间保存状态信息。当需要保存客户端与应用程序服务的交互状态时，就需要使用有状态会话 Bean。例如。网络购物中的购物车，就是一个有状态会话 Bean。用户通过客户端程序选购一些商品并放入购物车后，管理交互的有状态会话 Bean 必须保存这些选中的商品，直到删除商品或者付款购买商品。有状态会话 Bean 必须存储未处理的商品，并使用户能够添加更多商品。

会话 Bean 都必须创建一个接口，接口的类型有如下两种。

- 远程接口：选中“远程”复选框，将创建一个远程接口。该接口允许使用远程方法调用，从远程客户端访问会话 Bean。这通常是指没有在会话 Bean 的应用服务器上运行的客户端。与实体 Bean 和消息驱动 Bean 相比，会话 Bean 更有可能具有远程接口，因为它们通常是从远程客户端进行访问的入口点。如果客户端模块与 EJB 模块不在同一个企业应用程序中，则必须通过其远程接口来访问 Bean 的方法。
- 本地接口：选中“本地”复选框，将创建一个本地接口。该接口允许从相同 Java 虚拟机上运行的客户端访问会话 Bean。当在本机上创建 EJB，并创建客户端进行调试时，应该选择这种类型的接口。也就是说，如果客户端模块与 EJB 模块包含在同一个企业应用程序中，则这些模块可以通过其本地接口来访问会话 Bean 的模块。

这里选中“Stateless”单选按钮和“本地”复选框，单击“完成”按钮，创建会话 Bean。

Download at http://www.pin5i.com/

NetBeans 集成开发环境将创建一个本地接口和实现这个接口的 EJB 模块。本地接口的定义代码如下所示：

```
package myejb;

import javax.ejb.Local;

@Local
public interface MyEJBLocal {

}
```

这里引入了 javax.ejb.Local 类，并使用了@Local 声明。下面在接口定义中添加一个方法，实现一个简单的加法运算。

```
public interface MyEJBLocal {
    public int add(int num1, int num2);
}
```

自动创建的 EJB 代码如下所示，该模块实现了 MyEJBLocal 接口。注意，会话 bean 类和普通 Java 类的唯一不同之处就是多了@Stateless 声明。EJB 3.0 技术通过减少所需的代码量来简化会话 Bean 的创建。可以看到，标注@Stateless 用于将类声明为无态会话 Bean 组件，并且该类不再需要实现 javax.ejb.SessionBean 的语句。此外，代码也更为简捷了，因为利用 EJB 3.0 技术，业务方法不再需要使用代码来声明其抛出了所检查到的异常。

```
package myejb;

import javax.ejb.Stateless;

@Stateless
public class MyEJBBean implements MyEJBLocal {

    // Add business logic below. (Right-click in editor and choose
    // "EJB Methods > Add Business Method" or "Web Service > Add Operation")

}
```

下面添加 add 方法的实现代码：

```
public class MyEJBBean implements MyEJBLocal {

    // Add business logic below. (Right-click in editor and choose
    // "EJB Methods > Add Business Method" or "Web Service > Add Operation")
    public int add(int num1, int num2) {
        return num1+num2;
    }

}
```

以上是手工添加商业方法的过程，另外还可以利用 NetBeans 提供的可视化界面添加。在 Bean 源代码的任何位置单击鼠标右键，选择弹出菜单中的“EJB 方法”→“添加 Business 方法”命令，打开“添加 Business 方法”对话框，如图 15-4 所示。

在“名称”文本框中输入商业方法的名字，在“返回类型”文本框中选择或者输入方法的返回值类型。单击“添加”按钮添加一个参数。Name 列用于指定参数的名字；Type 列用于选择参数的类型。

至此，完成了一个简单的 EJB，该 Bean 提供了一个 add 方法，用于计算两个数的和。当然，在实际应用中，EJB 提供的功能应该更实用一些。这里只是为了演示目的，因此实现了一个简单的功能。

Download at http://www.pin5i.com/

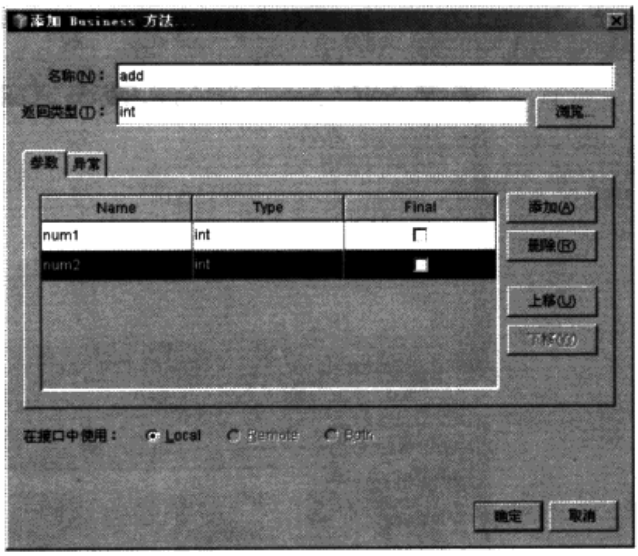


图 15-4 “添加 Business 方法”对话框

15.1.3 在 Web 应用程序中调用 EJB

在客户端，用户要通过 Web 应用程序使用 EJB 提供的服务，实现一些复杂的商业逻辑。当然对于用户来说，他并不知道什么时候调用了哪里的 EJB 模块。在位于应用层的 EJB 模块功能实现后，就可以在位于客户端的 Web 应用程序中调用这些 EJB 及其提供的服务（方法）了。

1. 在类路径中包含 EJB 项目

Web 应用程序项目必须在其类路径中包含 EJB 项目，以便调用 EJB。在“项目”窗口中，右键单击“SimpleEnterpriseAppDemo – war”项目节点，选择弹出菜单中的“属性”命令，打开“属性”对话框。单击“库”节点，确认 EJB 项目已经被添加到了 Web 项目的库路径中，否则请单击“添加项目”按钮进行手工添加。

2. JSF 受管 Bean

在项目属性对话框中，单击“框架”节点，然后单击“添加”按钮，打开“添加框架”对话框，如图 15-5 所示。

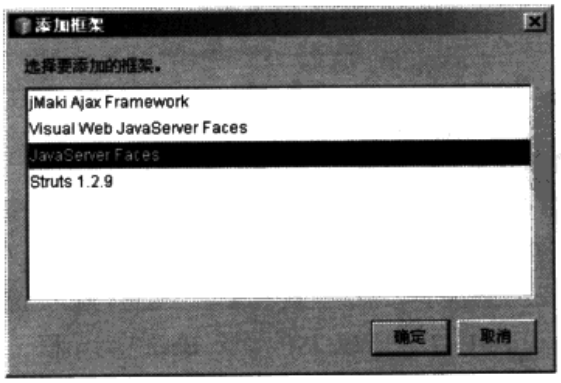


图 15-5 “添加框架”对话框

在这里有两个可以使用的框架：JSF 和 Struts。这里选中“JavaServer Faces”，然后单击“确定”按钮，添加 JSF 框架支持。添加 JavaServer Faces 框架后的“框架”页面如图 15-6 所示。

Download at http://www.pin5i.com/

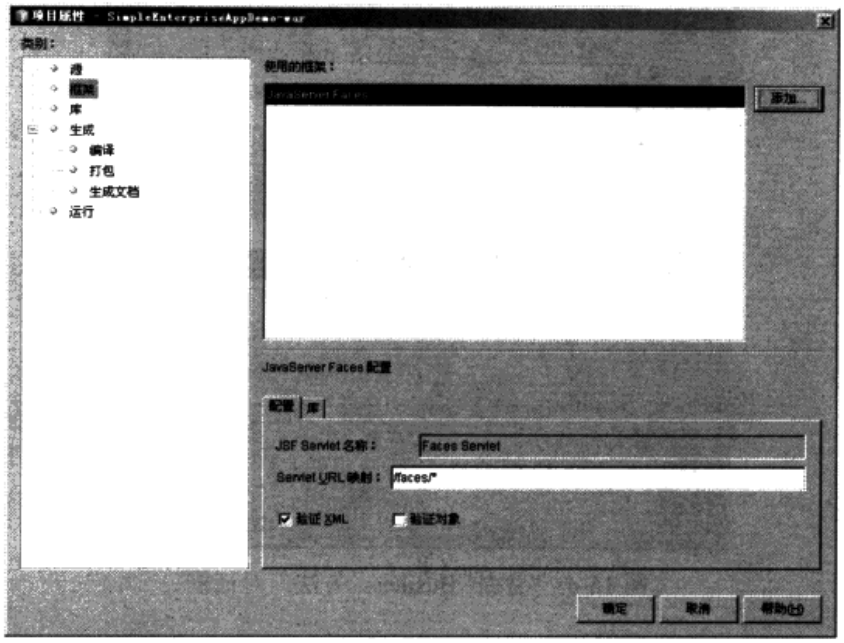


图 15-6 添加 JavaServer Faces 框架后的“框架”页面

这将创建文件 faces-config.xml，将所有库添加到项目的类路径中，注册相关条目到 web.xml 中，并创建一个 welcomeJSF.jsp 示例 JSF 页面。

在“项目”窗口中，右键单击“SimpleEnterpriseAppDemo – war”项目节点下的“源包”节点，选择弹出菜单中的“新建”→“文件/文件夹...”命令，打开“新建文件”对话框。在“类别”列表中选择“JavaServer Faces”节点，然后在“文件类型”列表中选择“JSF 受管 Bean”节点。单击“下一步”按钮，打开“新建 JSF 受管 Bean”对话框，如图 15-7 所示。

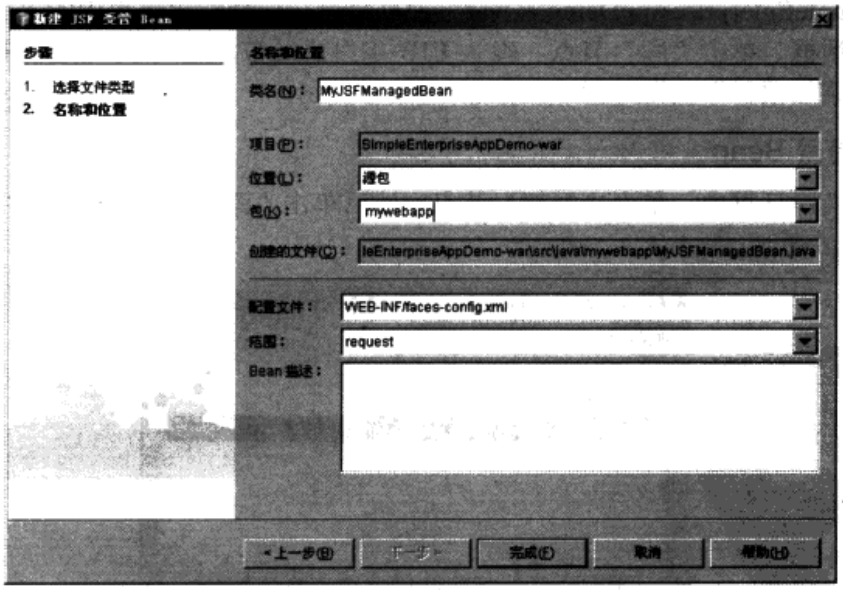


图 15-7 “新建 JSF 受管 Bean”对话框

JSF 受管 Bean 是由 JSF 实现管理的 JavaBean 组件。JSF 受管 Bean 描述了如何创建和管理 Bean。在“类名”文本框中输入 JSF 受管 Bean 的类名 MyJSFManagedBean。在“包”文本框中指定保存 JSF 受管 Bean 的包 mywebapp。

“配置文件”下拉列表框用于指定 JSF 配置文件。如果该文件不存在，则无法创建 JSF 受

Download at http://www.pin5i.com/

管 Bean。要快速而有效地创建 JSF 配置文件，请使用添加 JSF 框架的方法，让 NetBeans 自动创建该文件。

“范围”下拉列表框用于设置 JSF 受管 Bean 的范围，其可选值有以下几种。

- request：默认设置，即在当前请求范围内有效。
- session：在当前会话范围内有效。
- application：在整个应用程序范围内有效。
- none：无有效范围。

“Bean 描述”文本区域用于添加 JSF 受管 Bean 的描述信息。单击“完成“按钮，创建一个受管 Bean，代码如下所示：

```
package mywebapp;

public class MyJSFManagedBean {

    /** Creates a new instance of MyJSFManagedBean */
    public MyJSFManagedBean() {
    }

}
```

编写调用 EJB 的代码如下所示：

```
package mywebapp;

import myejb.MyEJBLocal;
import javax.ejb.EJB;

public class MyJSFManagedBean {

    /** Creates a new instance of MyJSFManagedBean */
    public MyJSFManagedBean() {
    }

    @EJB() MyEJBLocal myejb;
    public String getSum() {
        return String.valueOf(myejb.add(1,99));
    }

}
```

在 Java EE 5 中，开发人员不必编写任何 JNDI 查找代码，可以仅使用@EJB 声明来标识一个字段，服务器会在创建受管 bean 后自动注入一个 EJB 实例。

如果使用了 J2EE 1.4 规范，那么必须使用 Java 命名和目录服务接口 JNDI（Java Naming and Directory Interface）来查找 Bean 并检索本地或远程接口的实例。在每个 Enterprise Bean 的部署描述符中，为其指定了唯一的 JNDI 名称。假定 JNDI 的名字为“mybean.MyJSFManagedBean”，那么对应的代码为：

```
private MySessionRemote lookupMySessionBean() {
    try {
        Context c = new InitialContext();
        Object remote = c.lookup("mybean.MyJSFManagedBean");
        MyEJBRemoteHome rv =
            (MyEJBRemoteHome)PortableRemoteObject.narrow(remote,
                    MyEJBRemoteHome.class);
        return rv.create();
    } catch (…) {
    …
    }
}
```

Download at http://www.pin5i.com/

然后，必须使用类似于以下内容的代码来获取接口实例：

```
private MyEJBRemote newMyEJBRemote = lookupMySessionBean();
```

最后，在调用 Enterprise Bean 模块的部署描述符中注册 EJB 引用。是否添加引用以及添加引用的位置取决于是否符合下面的相应条件：

- 如果从 Enterprise Bean 实现类中进行查找，则将引用添加到对应的 Bean 中；
- 如果从 Web 应用程序的任何类中进行查找，则将引用添加到 Web 应用程序中；
- 如果从 EJB 模块的普通 Java 类中进行查找，则不添加任何引用。在这种情况下，必须手动将引用添加到调用普通 Java 类的 Bean 中。

3. 在页面中调用 EJB 模块

最后在 JSF 页面中，调用这个 JSF 受管 Bean，显示 EJB 的调用结果。在 welcomeJSF.jsp 页面中添加如下的代码：

```
<body>
    <f:view>
        <h1><h:outputText value="JavaServer Faces" /></h1>
        <h:outputText value="#{MyJSFManagedBean.sum}"/>
    </f:view>
</body>
```

打开 index.jsp 页面，添加如下的代码：

```
<a href="./faces/welcomeJSF.jsp">运行测试</a>
```

至此，完成了全部的企业应用程序的开发。运行企业应用程序项目，显示如图 15-8 所示的页面。

该页面为默认的 index.jsp 页面，单击“运行测试”超链接后，打开 welcomeJSF.jsp 页面，如图 15-9 所示。可以看到 EJB 被调用，返回了正确的计算结果。



图 15-8 index.jsp 页面



图 15-9 welcomeJSF.jsp 页面

15.1.4 企业应用程序的打包和部署

在默认情况下，通过部署包含 EJB 模块和 Web 应用程序模块的企业应用程序，可以同时部署这些模块。在“项目”窗口中，右键单击企业应用程序项目，然后可以选择以下弹出菜单命令。

- “运行”命令：将启动目标应用服务器和域，将企业应用程序部署到目标应用服务器和域中。如果应用程序已经部署过，则首先卸载企业应用程序。最后在 NetBeans 集成开发环境的默认浏览器中运行程序。
- “部署”命令：启动目标应用服务器和域，然后将企业应用程序部署到其中。

Download at http://www.pin5i.com/

如果要将 Java EE 应用程序部署到与应用程序开发服务器不同的服务器中，则可能需要对该应用程序进行一些调整。NetBeans 提供了“打包的归档”项目类型以简化此重新部署过程。

选择“新建”→“新建项目”命令，打开“新建项目”对话框。在“企业”列表中，选择“打包的归档”节点，然后单击“下一步”按钮，打开“创建新的可部署的归档项目”对话框，如图 15-10 所示。

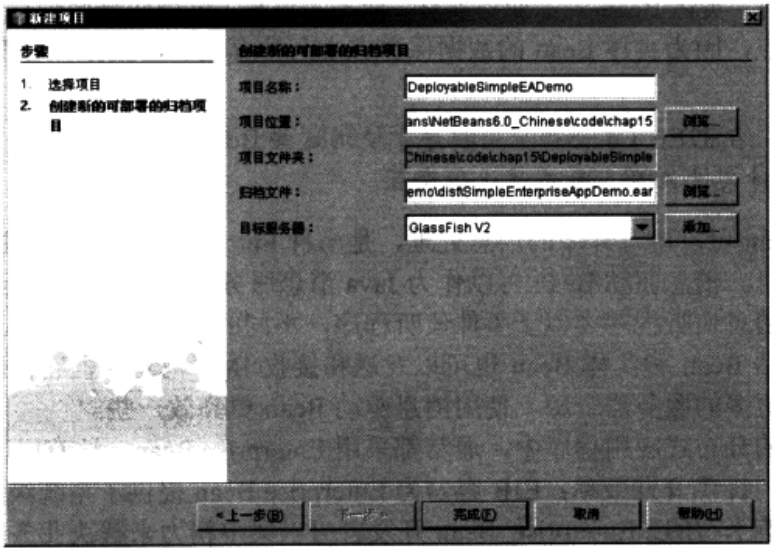


图 15-10 “创建新的可部署的归档项目”对话框

打包的归档项目基于现有的二进制归档文件，如 JAR 文件、WAR 文件或 EAR 文件。假定应用程序和服务器都符合 Java EE，则在重新部署该应用程序前不需要更改任何 Java 源代码。在“项目名称”文本框中输入归档项目的名字。单击“归档文件”文本框旁边的“浏览”按钮，选择以前的归档文件。单击“完成”按钮，完成项目的创建。

在“项目”窗口中，右键单击归档项目，选择弹出菜单中的“属性”命令，打开“项目属性”对话框。在“服务器”下拉列表框中可以选择重新部署的服务器类型。

另外，还要根据需要，添加、修改和删除归档文件的配置文件。例如，删除特定于服务器的部署描述符，并创建新的部署描述符，该部署描述符特定于要重新部署到的服务器。

配置完成后，右键单击项目节点，然后选择“生成”命令，生成项目。再右键单击项目节点，然后选择“验证项目”命令，验证更改的文件是否符合 Java EE 规范。最后，右键单击项目节点，然后选择“部署”命令，将应用程序部署到新的服务器上。

15.2 EJB 模块开发

企业 Java Bean（Enterprise Java Bean，EJB）是用 Java 编程语言编写的，主要用于封装应用程序业务逻辑的服务器端组件。业务逻辑是用于实现应用程序目的的代码。例如，在网络商城应用程序中，Enterprise Bean 可以在名为 checkOrderStatus 和 orderProduct 的方法中实现业务逻辑。通过调用这些方法，远程客户端可以检查订单的状态和定购商品。

15.2.1 Enterprise Bean 概述

在 NetBeans 集成开发环境中，可以开发三种类型的 Enterprise Bean：会话（Session）Bean、实体（Entity）Bean 和消息驱动（MessageDriven）Bean。

Download at http://www.pin5i.com/

会话 Bean 用于向特定用户会话提供服务，在会话需要时被创建。与实体 Bean 相比，会话 Bean 的生存期较短。会话 Bean 分为无状态会话 Bean 和有状态会话 Bean。关于这两者的区别以及创建无状态会话 Bean 的实例，已经在第 15.1.2 节给出。

实体 Bean 用于表示通常存储在数据库中的数据，是 EJB 2.1 规范的一部分。每个实体 Bean 都表示一个数据集，如数据库记录中的字段。实体 Bean 的每个实例都包含该数据的一个元素，如数据库表中的行。实体 Bean 表示业务实体，而不是过程（过程应该用会话 Bean）。实体 Bean 具有持久性的特点，因为实体 Bean 的数据保存在永久性存储中，例如保存在数据库中。



说明：在 J2EE 1.4 企业应用程序中使用这些 Bean。在 Java EE 5 企业应用程序中，应使用 Java 持久性 API 与数据库进行交互。

消息驱动 Bean 可以进行异步的消息处理，是一种 Enterprise Bean，它使得 J2EE 应用程序能够异步处理消息。消息驱动 Bean 可以作为 Java 消息服务 JMS（Java Message Service）的消息侦听程序，此消息侦听程序类似于事件侦听程序，不同的是消息驱动 Bean 接收消息而不是事件。其实，会话 Bean 和实体 Bean 也可以发送和接收 JMS 消息，但它们以同步方式运行。如果要避免占用更多的服务器资源，使用消息驱动 Bean 更高效一些。

在一些大型的分布式应用程序中，通常都采用 Enterprise Bean，这有以下几方面的原因。

- 简化开发，提供开发效率：EJB 容器为 Enterprise Bean 提供了系统级服务（如事务管理和安全授权），所以 Enterprise Bean 开发者可以集中精力来解决业务问题。对于客户端的开发者，由于 Enterprise Bean 包含了应用程序的业务逻辑，所有他们能够将精力集中于客户端表示上，而不必对实现业务规则或访问数据库的例程进行编码。这使得一些大型项目可以在不同的层次上进行并行开发，提高了工作效率。
- 良好的可维护性：对于业务逻辑的改变，只需要修改 Enterprise Bean 的实现代码即可，客户端的程序基本上不需要修改，这大大提高了系统的可维护性。
- 可移植：Enterprise Bean 是可移植的组件，所以应用程序组装程序可以通过现有 Bean 生成新的应用程序。如果这些应用程序都使用了标准 API，那么它们就可以在符合 Java EE 规范的任何服务器上运行。



15.2.2 开发实体 Bean

本节将通过一个实例演示如何创建实体 Bean，以及如何创建实体 Bean 的会话 Bean。创建一个名字为 CMPBeanDemo 的企业应用程序，并设置为主项目。

1．通过数据库创建实体 Bean

在“项目”窗口中单击“EJB”项目节点，选择弹出菜单中的“新建”→“通过数据库生成实体类”命令，打开“新建 通过数据库生成实体类”对话框，如图 15-11 所示。该对话框将通过现有的数据库连接来生成一组实体类。

选中“数据源”单选按钮，然后从后面的下拉列表框中选择数据源。如果还没有创建数据源，可以选择“新建数据源”选项，打开“创建数据源”对话框，如图 15-12 所示。

在“JNDI 名称”文本框中输入 JNDI 名字（必填），以便于进行查找定位。在“数据库连接”下拉列表框中，选择一个数据库连接。



说明：关于数据库的注册和创建数据库连接的方法，请读者参考本书第 7 章。

Download at http://www.pin5i.com/

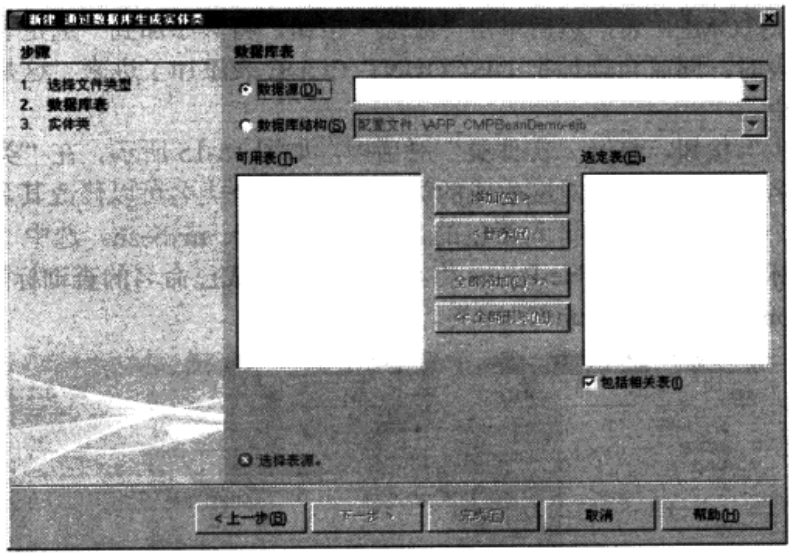


图 15-11 “新建 通过数据库生成实体类”对话框

这里，选择 jdbc/sample 数据源。如果系统需要登录，将会打开“连接”对话框，如图 15-13 所示。在输入正确的用户名和密码后，单击“确定”按钮，开始连接数据库。连接后，所有可以使用的数据表将显示在“可用表”列表中，如图 15-14 所示。

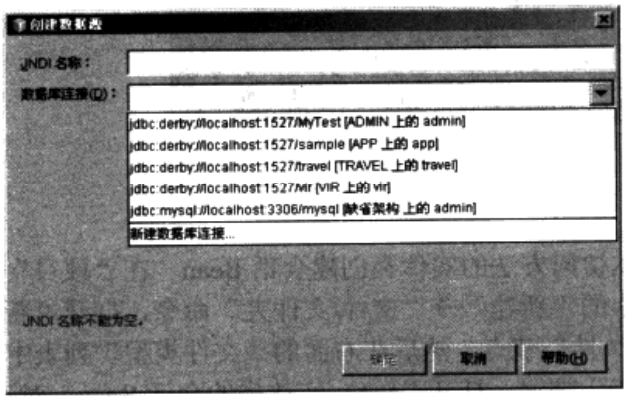


图 15-12 “创建数据源”对话框

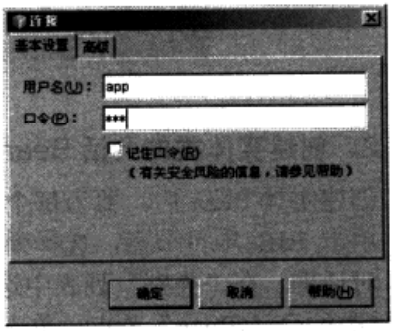


图 15-13 “连接”对话框

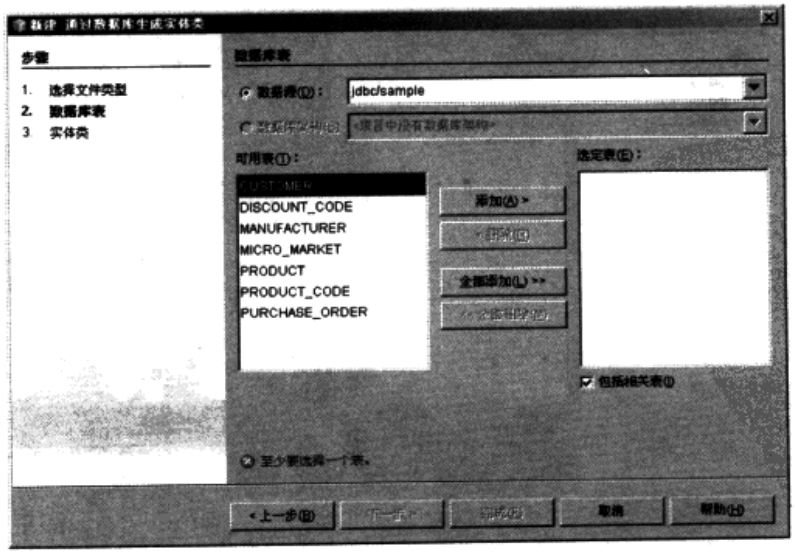


图 15-14 显示数据库中所有可以使用的表

Download at http://www.pin5i.com/

这里选中 CUSTOMER 表，然后单击“添加”按钮，将其添加到“选定表”列表中。在添加 CUSTOMER 表时，同时自动地把 DISCOUNT_CODE 表添加了进来，这是因为选中了“包括相关表”复选框。

单击“下一步”按钮，打开“实体类”对话框，如图 15-15 所示。在“类名称”表中给出了现有的数据表名字，以及将要对应生成的类名。双击某个类名可以修改其名字。从“包”下拉菜单中选择一个现有包，或者键入新包的名称，这里输入 mybean。选中“为持久性字段生成已命名的查询标注”复选框，NetBeans 将在实体类中生成已命名的查询标注。单击“完成”按钮，创建 Customer 类和 DiscountCode 类。

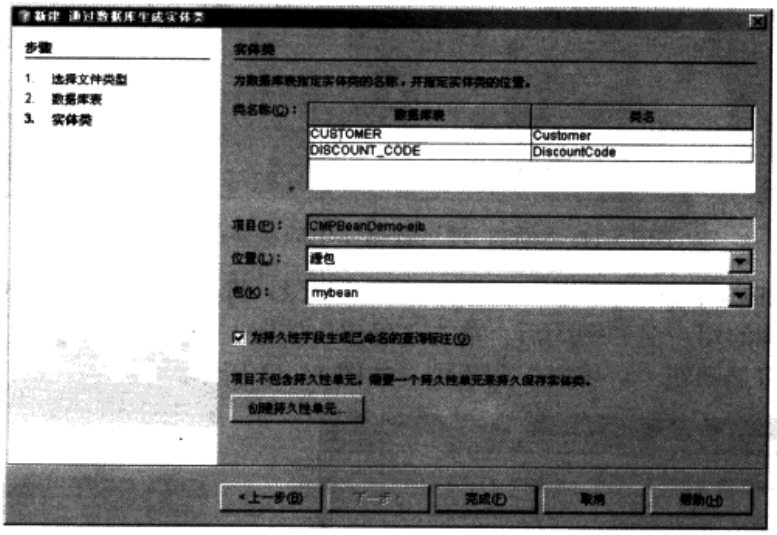


图 15-15 “实体类”对话框

2. 创建实体类的会话 Bean

创建实体 Bean 后，要为每个具有基本访问方法的实体类创建会话 Bean。在“项目”窗口中，单击 EJB 项目节点，选择弹出菜单中的“新建”→“文件/文件夹”命令，打开“新建文件”对话框。在“类别”列表中选择“持久性”节点，然后在右面的“文件类型”列表中选择“实体类的会话 Bean”节点。单击“下一步”按钮，打开“新建 实体类的会话 Bean”对话框，如图 15-16 所示。

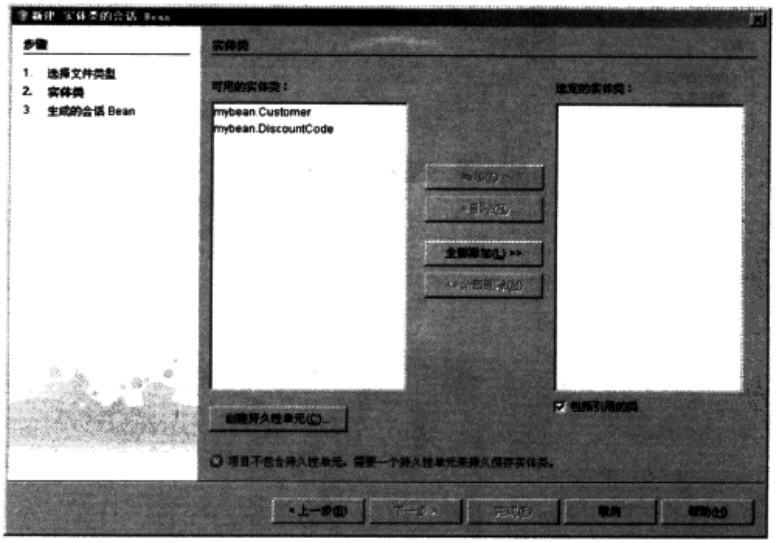


图 15-16 “新建 实体类的会话 Bean”对话框

Download at http://www.pin5i.com/

单击“创建持久性单元”按钮，打开“创建持久性单元”对话框。关于持久性单元，将在第 15.3 节中进行介绍，这里在“数据源”下拉列表框中选择 jdbc/sample，其他接受默认设置，单击“创建”按钮创建持久性单元。

在“可用的实体类”列表中选择所有的实体类，然后单击“添加”按钮，将它们添加到“选定的实体类”列表中。单击“下一步”按钮，打开“生成的会话 Bean”对话框，如图 15-17 所示。

如果选中“远程”复选框，则会自动为每个实体类创建 Façade 会话 Bean 和远程接口，即<类名>Façade 类和<类名>FacadeRemote 类。例如，对于 Customer 实体类，这里将创建 CustomerFacade 类和 CustomerFacadeRemote 类。

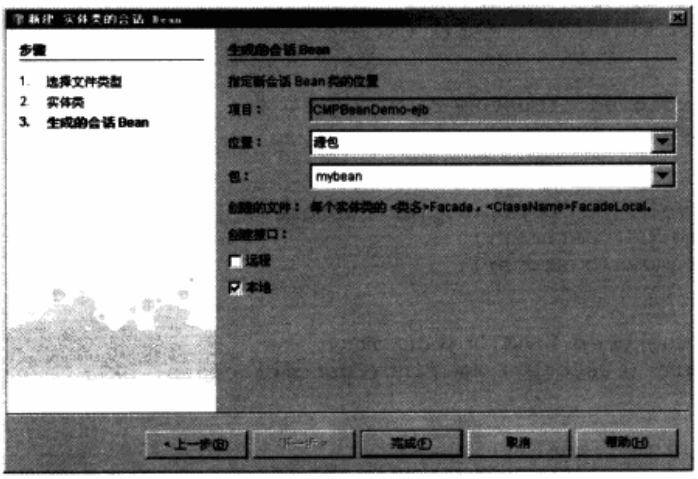


图 15-17 “生成的会话 Bean”对话框

如果选中“本地”复选框，则会自动为每个实体类创建 Facade 会话 Bean 和本地接口，即<类名>Facade 类和<类名>FacadeLocal 类。例如，对于 Customer 实体类，这里将创建 CustomerFacade 类和 CustomerFacadeLocal 类。这里仅选中“本地”复选框，单击“完成”按钮，创建会话 Bean 及其本地接口。下面是 CustomerFacadeLocal 的定义。在本地或者远程接口中，使用 create()方法创建一个 EJB 2.1 实体 Bean 主对象或本地主对象。remove()方法用于移除一个 EJB 2.1 实体 Bean 的实例。

```
package mybean;

import java.util.List;
import javax.ejb.Local;

@Local
public interface CustomerFacadeLocal {
    void create(Customer customer);

    void edit(Customer customer);

    void destroy(Customer customer);

    Customer find(Object pk);

    List findAll();
}
```

下面是 CustomerFacade 的实现代码。

```
package mybean;

import java.util.List;
```

Download at http://www.pin5i.com/


```
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;

@Stateless
public class CustomerFacade implements CustomerFacadeLocal {

    @PersistenceContext
    private EntityManager em;

    /** Creates a new instance of CustomerFacade */
    public CustomerFacade() {
    }

    public void create(Customer customer) {
        em.persist(customer);
    }

    public void edit(Customer customer) {
        em.merge(customer);
    }

    public void destroy(Customer customer) {
        em.merge(customer);
        em.remove(customer);
    }

    public Customer find(Object pk) {
        return (Customer) em.find(Customer.class, pk);
    }

    public List findAll() {
        return em.createQuery("select object(o) from Customer as o").getResultList();
    }

}
```

其中 EntityManager 类为实体管理器，用于管理所有 EJB 3.0 运行环境中的所有实体 Bean。EJB 3.0 通过 javax.persistence.EntityManager 类实现了持久性、查找和移除。在 EJB 3.0 实体 Bean 的客户类中，可使用@PersistenceContext 注释动态注入 EntityManager，即如下所示的代码：

```
@PersistenceContext
private EntityManager em;
```

表 15-1 是几个关于实体管理器类的代码说明。

表 15-1 关于实体管理器类的代码说明

代 码	说 明
em.persist(customer)	持久化使一个实体 Bean 实例 customer
em.merge(customer)	把指定的实体 Bean 实例 customer 合并到当前持久性上下文中
em.remove(customer)	移除实体 Bean 的一个实例 customer
em.find(Customer.class, pk)	查找一个实体 Bean 实例
em.createQuery("select object(o) from Customer as o").getResultList()	创建一个 Query 对象的实例，以执行命名查询。通过 getResultList()方法返回查询结果

3. 在 Web 客户端中显示数据

在创建了实体 Bean 及其会话 Bean 后，就可以在客户端程序中通过会话 Bean 来显示实体 Bean 中的数据了。在 CMPBeanDemo-war 项目中，创建一个名为 ListCustomers、包名为 myweb 的 Servlet。

Download at http://www.pin5i.com/

在类的开始通过@EJB 注入会话 Bean，代码如下所示：

```
@EJB
private CustomerFacadeLocal customerFacade;
```

然后在 processRequest()方法中，添加如下的代码：

```
protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Servlet ListCustomers</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h1>Servlet ListCustomers at " + request.getContextPath() + "</h1>");

    out.println("<table border=1>");
    out.println("<thead>" +
            "<tr><th>姓名</th><th>城市</th></tr></thead><tbody>");

    List list = customerFacade.findAll();
    for (Iterator it = list.iterator(); it.hasNext();) {
        Customer elem = (Customer) it.next();
        out.println("<tr>");
        out.println("<td>" + elem.getName()+ "</td>");
        out.println("<td>" + elem.getCity()+ "</td>");
        out.println("</tr>");
    }

    out.println("</tbody></table>");
    out.println("</body>");
    out.println("</html>");

    out.close();
}
```

其中，语句 customerFacade.findAll()用于返回所有的实体实例（每个实体实例，可以看作数据表的一行），然后再通过实体的 getName()方法返回客户的姓名，通过 getCity()方法返回客户所在的城市。

在添加必要的引用后，就可以运行应用程序了。在 IE 浏览器的地址栏中输入 Serlvet 的地址，显示如图 15-18 所示的界面。可以看到，表中列出了全部的客户姓名和所在城市。



图 15-18　程序运行界面

Download at http://www.pin5i.com/

图 15-19 是对数据库直接执行 SQL 查询的结果。通过对比可以看到，两个结果完成一致。

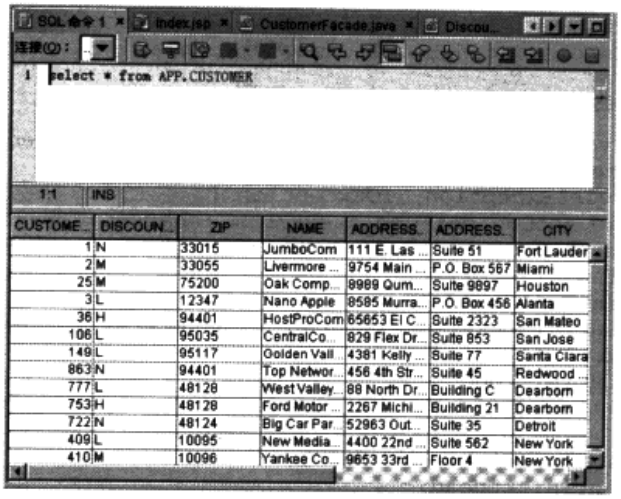


图 15-19 数据库中的客户信息

15.2.3 开发驱动消息 Bean

Java 消息服务（Java Message Service）一种与厂商无关的消息收发系统。JMS 能够通过消息收发服务从一个 JMS 客户端向另一个 JMS 客户短发送消息。JMS 的消息是异步的，也就是说，JMS 客户端可以发送消息而不必等待回应，因此可以通过消息驱动 Bean 来进行接收处理。下面将创建一个简单的消息驱动 Bean，打印客户端发送过来的消息。

创建一个名为 MessageBeanDemo 企业应用程序，并设置为主项目。在“项目”窗口中，右键单击 EJB 模块节点（这里为 MessageBeanDemo-ejb 节点），选择弹出菜单中的“新建”→“消息驱动 Bean”命令。打开“新建 消息驱动 Bean”对话框，用来创建新的消息驱动 Bean，如图 15-20 所示。

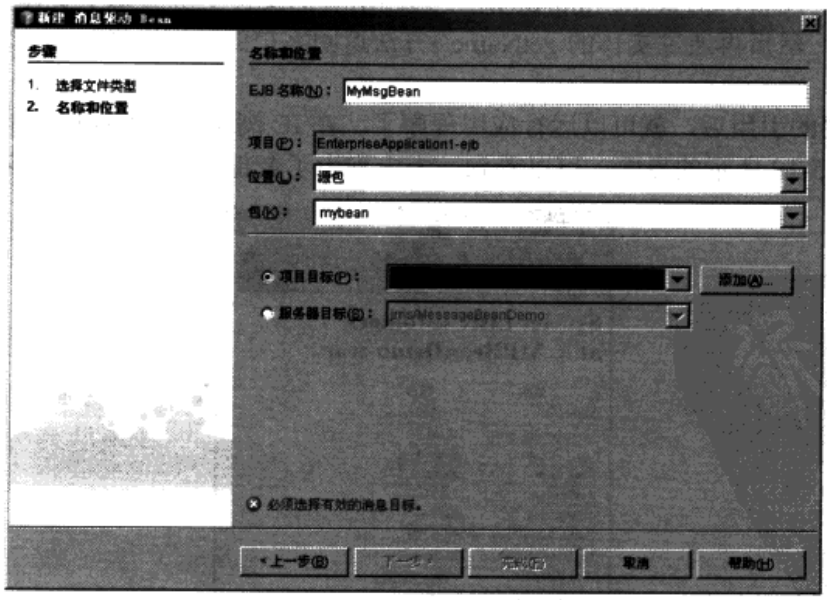


图 15-20 “新建 消息驱动 Bean”对话框

在“EJB 名称”文本框中输入 Bean 的名称 MyMsgBean，在“包”下拉列表框中选择现有的包或者输入新包的名称 mybean。

Download at http://www.pin5i.com/

选中“项目目标”单选按钮，单击后面的“添加”按钮，打开“添加消息目标”对话框，如图 15-21 所示。

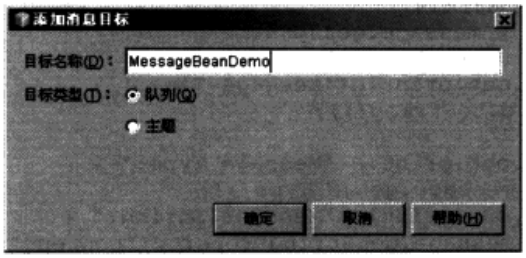


图 15-21 “添加消息目标”对话框

在“目标名称”文本框中输入 MessageBeanDemo。消息驱动 Bean 的目标类型有以下两个选项：

- “队列”单选按钮：使用点对点消息传送域，并且最多只有一个使用者。
- “主题”单选按钮：使用发布—订阅消息传送域。具有此目标类型的 Bean 可能有零个、一个或多个使用者。

这里，选中“队列”单选按钮，单击“确定”按钮，返回“新建 消息驱动 Bean”对话框。单击“完成”按钮，自动创建如下的代码：

```
package mybean;

import javax.ejb.ActivationConfigProperty;
import javax.ejb.MessageDriven;
import javax.jms.Message;
import javax.jms.MessageListener;

@MessageDriven(mappedName = "jms/MessageBeanDemo", activationConfig =  {
        @ActivationConfigProperty(propertyName = "acknowledgeMode",
propertyValue = "Auto-acknowledge"),
        @ActivationConfigProperty(propertyName = "destinationType",
propertyValue = "javax.jms.Queue")
    })
public class MessageBeanDemoBean implements MessageListener {

    public MessageBeanDemoBean() {
    }

    public void onMessage(Message message) {
    }

}
```

语句@MessageDriven(mappedName = "jms/ MessageBeanDemo "使用标注向容器说明，该组件是一个消息驱动 Bean 并且该 Bean 使用 JMS 资源。当 NetBeans 集成开发环境生成类时，将从类名称（MyMsgBean.java）派生资源的映射名（jms/MyMsgBean）。JMS 资源被映射到目标的 JNDI 名称，Bean 从该目标中接收消息。

使用 NetBeans 提供的“新建 消息驱动 Bean”对话框创建消息驱动 Bean 时，可以自动创建 JMS 资源。通过 EJB 3.0 API，可以从 Bean 类中查找 JNDI 名称空间中的对象，这样就不需要配置部署描述符来指定 JMS 资源了。

在 onMessage()方法中，添加如下的代码，打印接收到的消息。

```
public void onMessage(Message message) {
    try {
        System.out.println("Message ID:" +
                message.getJMSMessageID());
```

Download at http://www.pin5i.com/


```
            System.out.println("Message Expiration:" +
                    message.getJMSExpiration());

            //显示消息的时间
            long t = message.getJMSTimestamp();
            Date d = new Date(t);
            System.out.println("Message Time:" +
                    d.toString());

            System.out.println("Message Type:" +
                    message.getJMSType());
            System.out.println("Message String:" +
                    message.getStringProperty("content"));
        } catch (JMSException ex) {
            ex.printStackTrace();
        }
    }
```

getJMSMessageID()方法用于返回消息的 ID；getJMSTimestamp()方法返回消息的时间；message.getStringProperty("content")返回字符串类型的、关键字为 content 的值，该值可以由 getStringProperty()方法进行设置。

下面在客户端添加发送消息的代码。在 MessageBeanDemo-war 项目中，打开 index.jsp 文件，添加如下的代码：

```
<h1>发送消息</h1>
<form name="form" action="SendMsg">
    <input type="text" name="MsgContent" value="" />
    <input type="submit" value="发送" name="submit" />
</form>
```

创建一个名为 SendMsg 的 Servlet，包名字定义为 myweb。添加 import javax.jms.*引用，然后添加如下的代码。

```
@Resource(name = "jms/MessageBeanDemo")
private Queue messageBeanDemo;

@Resource(name = "jms/MessageBeanDemoFactory")
private ConnectionFactory messageBeanDemoFactory;
```

EJB 3.0 规范允许使用@Resource 标注将资源直接引入类中，这里引入了 JMS 连接工厂和队列。在 processRequest()方法中添加如下的代码，用于发送消息。

```
protected void processRequest(HttpServletRequest request, HttpServlet
                              Response response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");

    //获得用户输入的字符串
    String content =request.getParameter("MsgContent");
    if (content!=null) {
        try {
            //创建 JMS 连接
            Connection connection = connectionFactory.createConnection();
             Session session = connection.createSession(false, Session.AUTO_
                               ACKNOWLEDGE);
            MessageProducer messageProducer = session.createProducer(queue);

            //创建消息对象
            ObjectMessage message = session.createObjectMessage();
            //设置 setStringProperty 属性，关键字为 content
            message.setStringProperty("content", content);

            //发送消息
```

Download at http://www.pin5i.com/


```
                messageProducer.send(message);
                messageProducer.close();
                connection.close();

                //重定向到 index.jsp 文件
                response.sendRedirect("index.jsp");

            } catch (JMSException ex) {
                ex.printStackTrace();
            }
        }
    }
```

运行程序，显示界面如图 15-22 所示。输入任何的字符串以后，单击“发送”按钮，将执行 SendMsg Servlet 发送消息。消息 Bean 接收到消息后，会在 GlassFish 控制台上显示如下所示的信息：

```
Message ID:ID:8-192.168.1.65(fd:53:41:82:16:a0)-1-1205662259062
Message Expiration:0
Message Time:Sun Mar 16 03:10:59 PDT 2008
Message Type:null
Message String:Hello Message Bean
```



图 15-22 消息驱动 Bean 客户端界面

15.2.4 自动生成发送 JMS 消息的代码

在上一节中，手工添加了发送 JSM 消息的代码。本节以上一节的例子为基础，介绍如何自动生成发送代码。

在 SendMsg Servlet 中，右键单击空白区域。选择弹出菜单中的“企业资源”→“发送 JMS 消息”命令，打开“发送 JMS 消息”对话框，如图 15-23 所示。

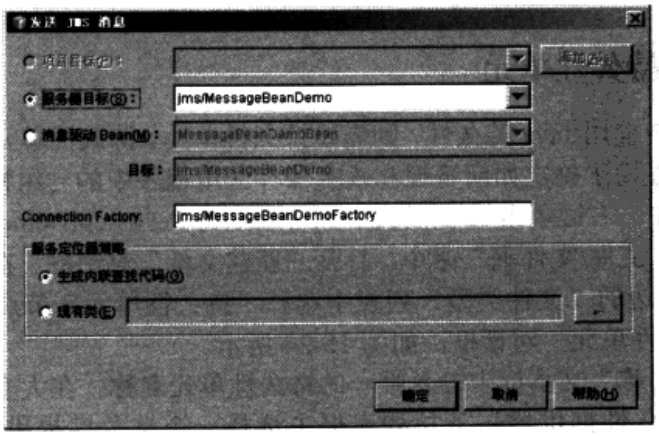


图 15-23 “发送 JMS 消息”对话框

Download at http://www.pin5i.com/

选中“服务器目标”单选按钮，在后面的下拉列表框中选择接收消息的 Bean。如果选中“生成内联查找代码”单选按钮，则使用内联查找代码。如果选中“现在类”单选按钮，则需要单击旁边的按钮来搜索服务定位器类名。单击“确定”按钮，NetBeans 将创建如下的代码：

```
            private Message createJMSMessageForjmsMessageBeanDemo(Session session,
Object messageData) throws JMSException {
        }

        private void sendJMSMessageToMessageBeanDemo(Object messageData) throws
JMSException {
            Connection connection = null;
            Session session = null;
            try {
                connection = messageBeanDemoFactory.createConnection();
                session = connection.createSession(false,
javax.jms.Session.AUTO_ACKNOWLEDGE);
                MessageProducer messageProducer =
session.createProducer(messageBeanDemo);
                messageProducer.send(createJMSMessageForjmsMessageBeanDemo(session,
messageData));
            } finally {
                if (session != null) {
                    session.close();
                }
                if (connection != null) {
                    connection.close();
                }
            }
        }
```

其中，createJMSMessageForMyMsgBean()方法用于创建发送的消息，应该在此方法中实现业务逻辑；sendJMSMessageToMyMsgBean()方法处理在上面的方法中创建的消息。上一节中的发送消息代码就在这个方法中被实现了，开发人员不需要修改此方法中的代码。

15.3 使用持久性 API

Java 持久性 API（JSR-220）是在 Java EE 5 平台引入的。该 API 不但可以在 Java EE 平台的 JB 组件外部使用（例如，在 Web 应用程序和应用程序客户端中使用），而且还可以在 Java EE 平台之外的 Java SE 应用程序中使用。

下面通过一个实例演示，如何创建持久性单元，以及如何使得持久性单元和实体类协同工作。首先创建一个名字为 PersistentDemo 的企业应用程序，并设置为主项目。

15.3.1 创建持久性单元

如果在应用程序中使用 Java 持久性，则需要持久性单元。持久性单元是一个具有唯一名称的属性集合，这些属性用于确定如何管理和永久保存应用程序中的一组特定实体。

在“项目”窗口中，右键单击 EJB 模块节点（这里为 PersistentDemo-ejb 节点），选择弹出菜单中的“新建”→“文件/文件夹”命令。打开“新建文件”对话框。在“类别”列表中选择“持久性”节点，在右边的“文件类型”列表中选择“持久性单元”节点。单击“下一步”按钮，打开“新建 持久性单元”对话框，如图 15-24 所示。

在“持久性单元名称”文本框中输入唯一的持久性单元名称。在大多数情况下，可以使用默认的名称。在“持久性提供器”下拉列表框中选择一个持久性提供程序或库（例如选择 Hibenate），或者选择“新建持久性库”以添加新库。NetBeans 集成开发环境默认使用 TopLink

Download at http://www.pin5i.com/

Essentials 持久性提供程序，它是 Sun Java System Application Server Platform Edition 9 和 GlassFish 中默认的 Java 持久性提供程序。TopLink Essentials 持久性提供程序包含在 TopLink Essential.jar 文件中，该文件还包含了实体管理器。

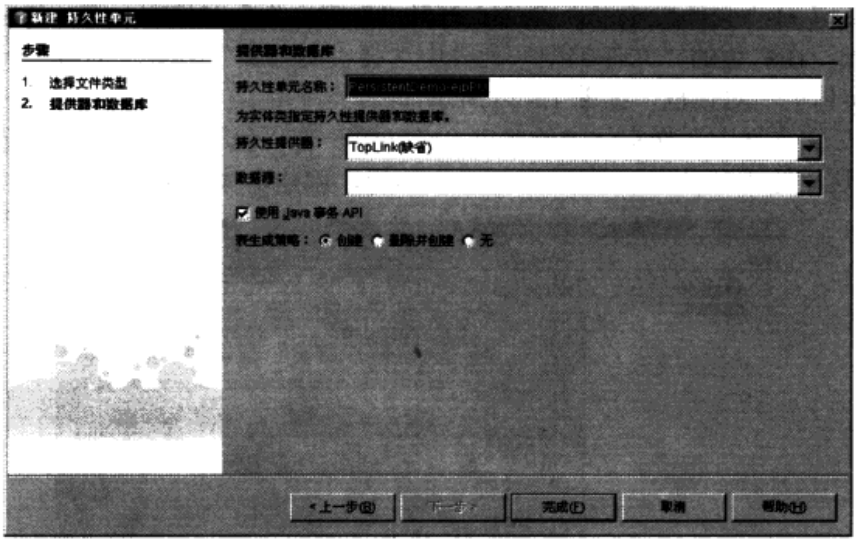


图 15-24 “新建 持久性单元”对话框

在“数据源”下拉列表框中选择存储持久性实体的数据库。选中“使用 Java 事务 API”复选框，通过 Java 事务 API 来进行事务处理。

表的生成策略有如下几种类型。

- “创建”单选按钮：在部署应用程序时创建基于实体类的表。
- “删除并创建”单选按钮：先删除表，然后再创建。
- “无”单选按钮：不进行任何操作。

单击“完成”按钮，创建 persistence.xml 文件。持久性单元就在 persistence.xml 文件中定义。在 persistence.xml 文件中定义多个持久性单元，但每个持久性单元必须具有唯一的名称。persistence.xml 文件可以在可视化的编辑器中进行编辑，如图 15-25 所示。

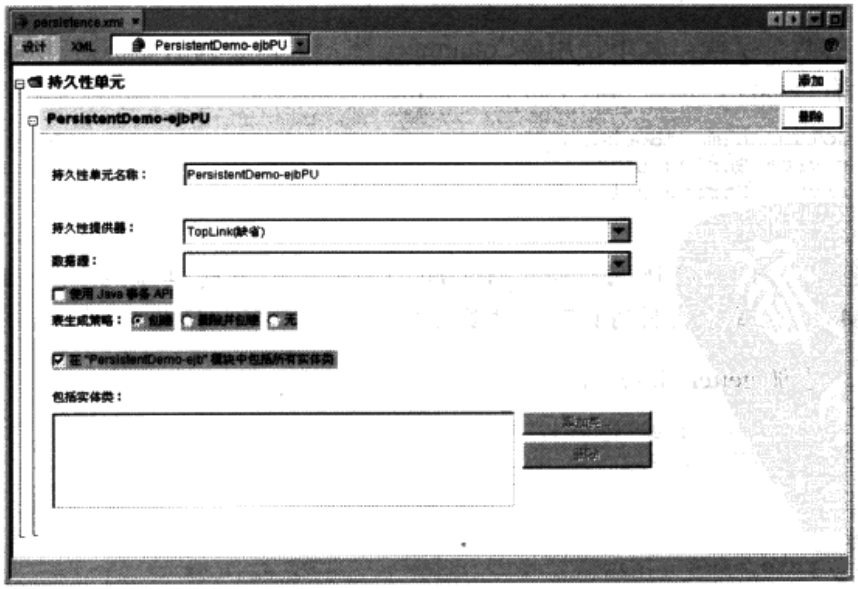


图 15-25 persistence.xml 文件的可视化编辑器

Download at http://www.pin5i.com/

15.3.2 创建实体 Bean

在“项目”窗口中，右键单击 EJB 模块节点（这里为 PersistentDemo-ejb 节点），选择弹出菜单中的“新建”→“文件/文件夹”命令。打开“新建文件”对话框。在“类别”列表中选择“持久性”节点，在右边的“文件类型”列表中选择“实体类”节点。单击“下一步”按钮，打开“新建 实体类”对话框，如图 15-26 所示。

在“类名”文本框中选择实体类的名字 BookInfo。在“包”下拉列表框中选择包或者输入新的包名字，这里输入 mybean。单击“完成”按钮，创建实体 Bean。

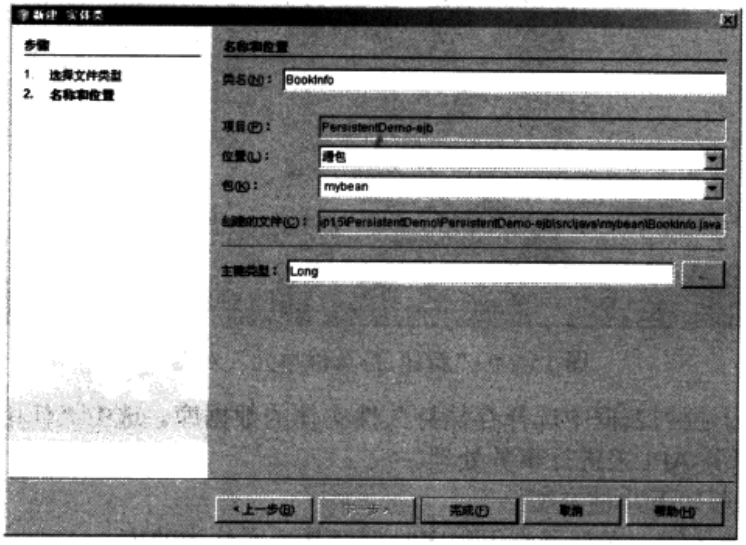


图 15-26 “新建 实体类”对话框

每个实体类都必须具有一个主键，因此在创建实体类时，NetBeans 会添加@Id 标注以声明要用作主键的字段。此外，NetBeans 还会添加@Generated 标注以指定主键的生成策略。

```
@Id
@GeneratedValue(strategy = GenerationType.AUTO)
private Long id;
```

添加如下的两个字段。其中使用@Column 标注，来声明列的名称是 BookName 而不是 name。如果没有该标注，那么列名字将会为 name。

```
@Column(name="BookName")
private String name;
private String author;
```



说明：还可以使用@ManyToOne 标注来表明列具有多对一的关系；使用@OneToMany 标注来表明列具有一对多的关系。

为两个字段添加 getter 和 setter 方法：

```
public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getAuthor() {
```

Download at http://www.pin5i.com/


```
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }
```

最后，创建一个实体类的会话 Bean，使用本地接口创建即可。

15.4 创建 GlassFish 资源

在 NetBeans 集成开发环境中，还可以创建一些 GlassFish 资源，例如 JDBC 连接池资源、持久性资源等，下面分别进行介绍。

15.4.1 JDBC 连接池资源

JDBC 连接池是一组可重复使用的连接，应用服务器为特定数据库维护这些连接。请求数据库连接的应用程序将从该池中获取连接。当应用程序关闭某个连接后，该连接将会返回到连接池中。

在 NetBeans 集成开发环境中，选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”列表中选择“Sun 资源”节点，然后在右边的“文件类型”列表中选择“JDBC 连接池”节点。

单击“下一步”按钮，打开“新建 JDBC 连接池”对话框，用于命名连接池并指定要使用的数据库连接，如图 15-27 所示。

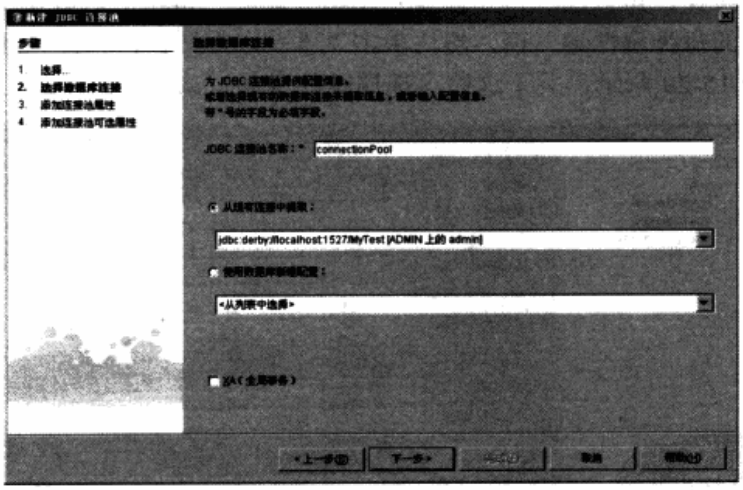


图 15-27 “新建 JDBC 连接池”对话框

在“JDBC 连接池名称”文本框中，输入连接池使用的名字。如果选中“从现有连接中提取”单选按钮，可以在下面的下拉列表框中选择使用已经存在的数据库连接。如果选中“使用数据库新建配置”单选按钮，那么可以从下面包含已知 JDBC 驱动程序的数据库列表中选择一种驱动程序。或者，选择“其他”选项，使用未列出的数据源。

如果使用 XA 事务，请选择“XA（全局事务）”单选按钮。单击“下一步”按钮，打开“添加连接池属性”对话框，如图 15-28 所示。

在“数据源类名”文本框中给出了数据源的类名，即实现数据源 API 的 JDBC 驱动程序。此名称是根据上一个屏幕中选定的数据库连接提供的。如果在上一个屏幕中选择“其他”作为

Download at http://www.pin5i.com/

连接，则可能需要手动添加数据源类名。

“资源类型”文本框中显示了连接池是允许 XA 全局事务，还是仅限于本地事务。可以返回到上一个对话框更改资源类型。

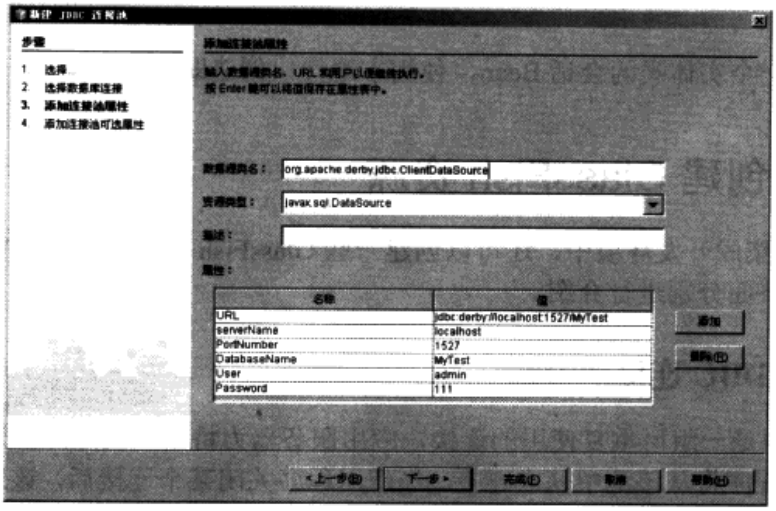


图 15-28 “添加连接池属性”对话框

在“属性”列表中给出了连接的具体属性。连接池属性可能会随数据库供应商的不同而有所变化。常用属性包括数据源 URL、用户名和口令等。根据数据库连接的不同可能会有其他建议或要求的属性。如果在图 15-27 所示的对话框中选择了一个现有数据连接，则该连接的一些属性和属性值，则可能会自动列在“属性”列表中。

单击“添加”按钮，可以添加一个属性；单击“删除”按钮，可以删除当前选中的属性。为“属性”列表中的每种属性输入值，然后单击“下一步”按钮，打开“添加连接池的可选属性”对话框，如图 15-29 所示。关于这些可选属性的说明，请参考表 15-2。

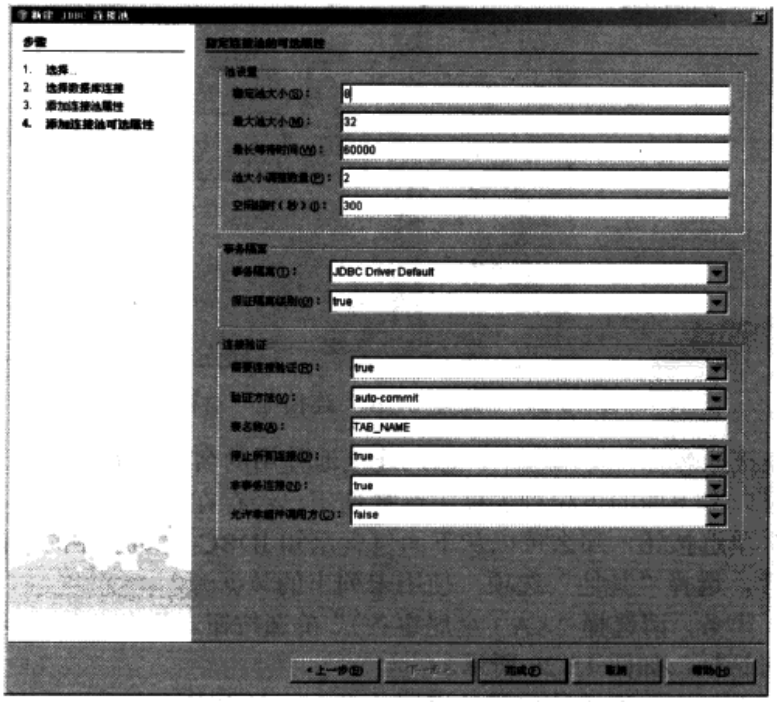


图 15-29 “添加连接池的可选属性”对话框

Download at http://www.pin5i.com/

表 15-2 JDBC 连接池可选属性

可选属性	说 明
稳定池大小	一个整数值，表示任意给定时刻池中的连接数。通常，这是池中的最小连接数
最大池大小	一个整数值，表示池中的最大连接数
最长等待时间	一个整数值，表示最长等待时间（以毫秒为单位）
池大小调整数量	在连接池收缩至最小池大小时，分批调整其大小。此数字表示池大小调整批次中的连接数。如果批量过大，可能会延迟连接的回收时间；如果批量过小，则可能会降低效率
空闲超时（秒）	一个整数值，表示最大空闲超时长度（以秒为单位）。这是连接可在池中处于空闲状态的时间。如果连接处于空闲状态的时间超过此时间，则会将其关闭
事务隔离	可以从下拉列表框中选择如下几种属性值。 • read-uncommitted：读取未提交数据 • read-committed：读取已提交数据 • repeatable-read：可重复读取 • serializable：可序列化 • JDBC Driver Default：JDBC 驱动的默认值
保证隔离级别	如果设置为 true，从池中提取的所有连接具有相同的隔离级别
需要连接验证	如果设置为 true，在连接发送到应用程序之前，应用服务器将对连接进行验证。当验证生效时，应用服务器可以自动恢复由于网络或数据库服务器故障而中断的数据库连接
验证方法	有如下几种验证方式。 • auto-commit：自动提交，应用服务器通过调用 con.getAutoCommit()方法来验证连接 • meta-data：元数据，应用服务器通过调用 con.getMetaData()方法来验证连接 • table：表，应用程序使用 select (count *) from <table-name>查询语句来查询指定的数据库表
表名称	这是连接验证表的名称。如果选择表作为验证方法，则需要设置表名称
停止所有连接	如果设置为 true，并且单个连接失败，应用服务器将关闭池中的所有连接，然后重新建立这些连接。如果设置为 false，则只会重新建立个别连接
非事务连接	如果设置为 true，则使用非事务连接
允许非组件调用方	如果设置为 true，则允许使用非组件调用方

设置完毕后，单击“完成”按钮，即可创建一个 JDBC 连接池资源。创建的资源可以在“项目”窗口中，通过打开项目节点下面的“服务器资源”节点进行浏览。



15.4.2 JDBC 资源

在 NetBeans 集成开发环境中，选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”列表中选择“Sun 资源”节点，然后在右边的“文件类型”列表中选择“JDBC 资源”节点。

单击“下一步”按钮，打开“新建 JDBC 资源”对话框，如图 15-30 所示。选中“使用现有 JDBC 连接池”单选按钮，从下面的下拉列表框中选择一个已经创建好的 JDBC 连接池。如果选中“创建新的 JDBC 连接池”单选按钮，那么会在后续步骤中增加创建 JDBC 连接池的步骤（具体每个步骤的说明，请参考上一小节的介绍）。

在“JNDI 名称”文本框中输入资源的 JNDI 名称（必填内容）。在“对象类型”下拉列表框中，选择对象的类型。如果在“启用”下拉列表框中选择“true”，那么将允许启用资源；设置为“false”，则可禁用资源。

Download at http://www.pin5i.com/

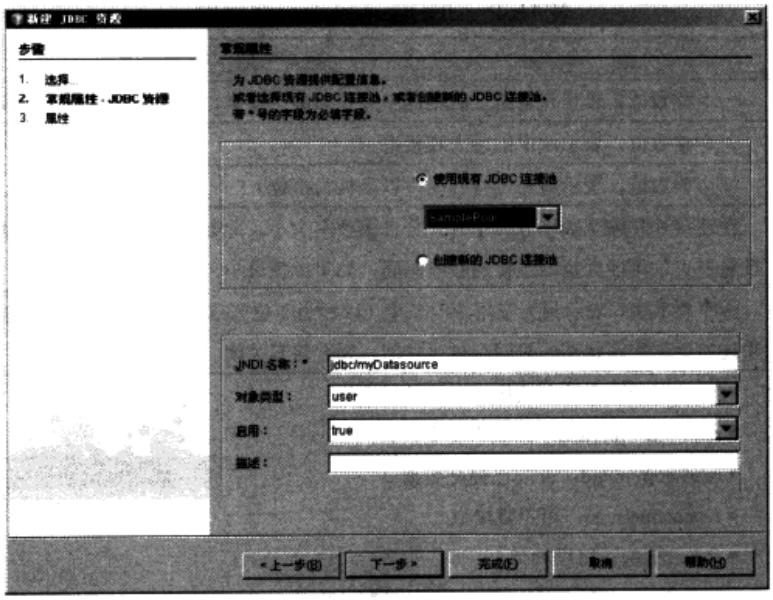


图 15-30 “新建 JDBC 资源”对话框

单击“下一步”按钮，打开“附加属性”对话框，如图 15-31 所示。单击“添加”按钮，可以为 JDBC 资源添加额外的配置信息和属性。

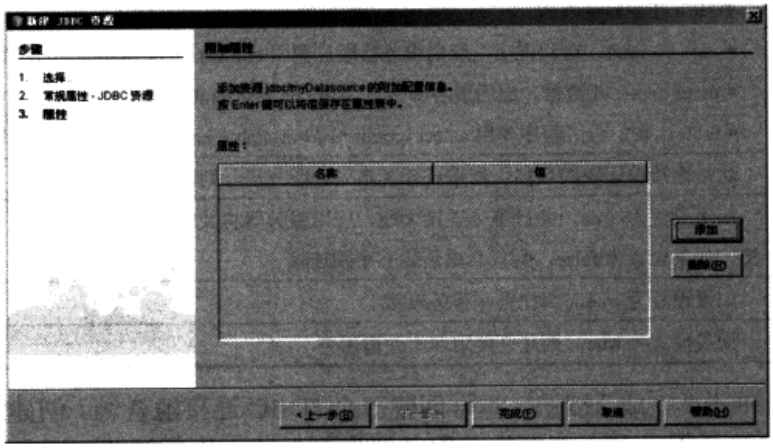


图 15-31 “附加属性”对话框

单击“完成”按钮，创建一个 JDBC 资源。创建的资源可以在“项目”窗口中，通过打开项目节点下面的“服务器资源”节点进行浏览。



15.4.3 JMS 资源

在 NetBeans 集成开发环境中，选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”列表中选择“Sun 资源”节点，然后在右边的“文件类型”列表中选择“JMS 资源”节点。

单击“下一步”按钮，打开“新建 JMS 资源”对话框，如图 15-32 所示。该对话框可以选择创建管理对象资源（目标）或连接器资源（连接工厂），以及指定资源的 JNDI 名称。

在“JNDI 名称”文本框中输入唯一的 JNDI 名称（必填属性）。所有资源 JNDI 名称都位于 java:comp/env 子上下文中，所以在指定 JMS 资源的 JNDI 名称时，只需输入 jms/name。

Download at http://www.pin5i.com/

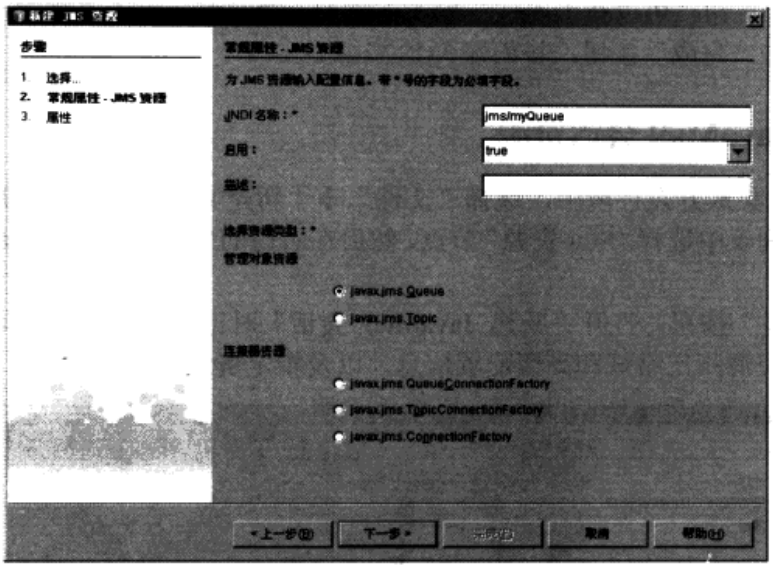


图 15-32 “新建 JMS 资源”对话框

注意：在服务器中注册资源后，无法再更改资源的 JNDI 名称。要更改 JNDI 名称，必须先将该资源从服务器中删除，然后在资源的“属性”窗口中更改 JNDI 名称并再次注册资源。

如果在“启用”下拉列表框中，选择“true”，则会启用资源，否则禁用资源。JMS 资源可以是管理对象资源，也可以是连接器资源。如果创建的 JMS 资源为管理对象资源，那么可以选择如下的选项。

- javax.jms.Queue 单选按钮：是队列。
- javax.jms.Topic 单选按钮：是主题。

如果创建的 JMS 资源为连接器资源，那么可以选择如下的选项。

- javax.jms.QueueConnectionFactory 单选按钮：被用作队列。
- javax.jms.TopicConnectionFactory 单选按钮：被用作主题。
- javax.jms.ConnectionFactory 单选按钮：被用作连接工厂。

单击“下一步”按钮，打开“JMS 属性”对话框，为 JMS 资源指定附加属性，或删除先前添加的属性，如图 15-33 所示。

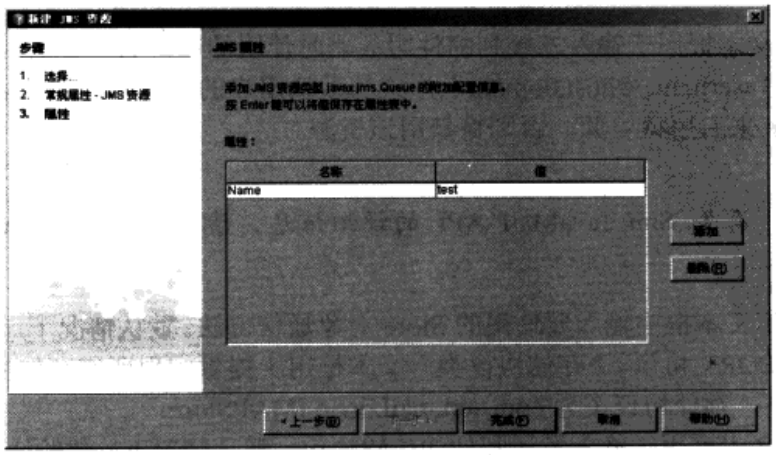


图 15-33 “JMS 属性”对话框

Download at http://www.pin5i.com/

单击“添加”按钮，可以在“属性”列表中添加一个属性；单击“删除”按钮，可以删除选中的属性。单击“完成”按钮，将创建 JMS 资源。

15.4.4 JavaMail 会话资源

在 NetBeans 集成开发环境中，选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”列表中选择“Sun 资源”节点，然后在右边的“文件类型”列表中选择“JavaMail 会话”节点。

单击“下一步”按钮，打开“新建 JavaMail 会话”对话框，如图 15-34 所示。该对话框为 JavaMail 会话资源指定常规和高级配置信息，以及指定资源的 JNDI 名称。

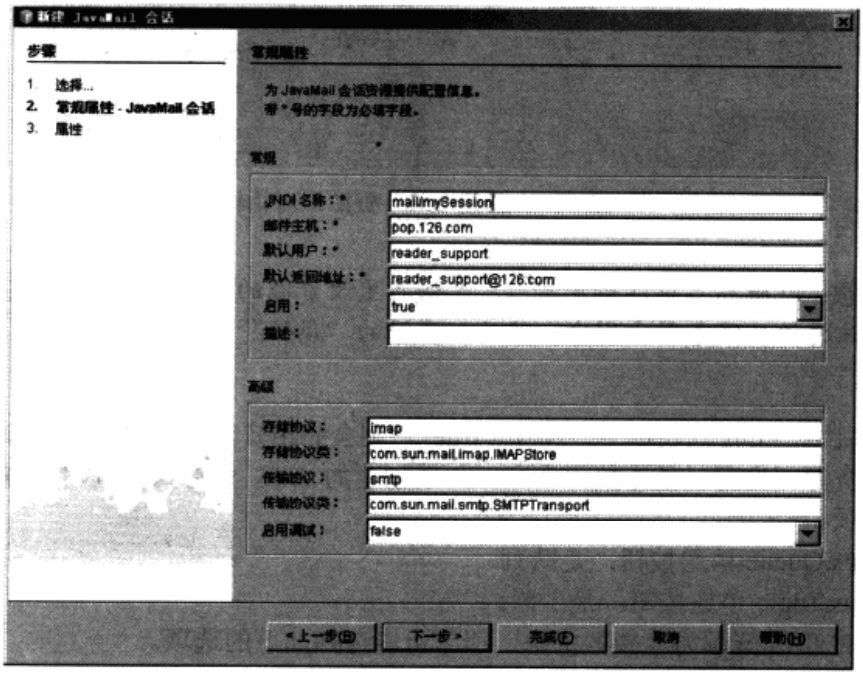


图 15-34 “新建 JavaMail 资源”对话框

在“JNDI 名称”文本框中输入资源的唯一的 JNDI 名称（必填）。JavaMail 资源通常放在 mail 命名子上下文中。

“邮件主机”文本框用于指定默认邮件服务器的 DNS 名称（必填）。此名称必须能够解析为实际的主机名称。

“默认用户”文本框用于输入连接到邮件服务器时使用的用户名（必填）。“默认返回地址”文本框用于输入“username@host.domain”形式的默认用户的电子邮件地址。在“启用”下拉列表框中选择 true 来启用该资源，否则将禁用该资源。

说明：有关 Sun JavaMail API 的详细信息，请参见 http://java.sun.com/products/javamail/。

在“存储协议”文本框中输入要使用的 Store 对象通信方法。默认情况下，存储协议为 imap，用户也可以使用 POP3 协议。“存储协议类”文本框用于提供实现所需存储协议的存储通信方法类。默认情况下，存储协议类为 com.sun.mail.imap.IMAPStore。

“传输协议”文本框用于输入发送邮件的传输协议。默认情况下，传输协议为 smtp。“传输协议类”文本框用于定义实现传输协议的方法类。默认情况下，传输协议类为 com.sun.mail.smtp.

Download at http://www.pin5i.com/

SMTPTransport。

在“启用调试”下拉列表框中选择 true，将启用此邮件会话的调试输出。单击“下一步”按钮，打开“邮件会话属性”对话框，如图 15-35 所示。

单击“添加”按钮，添加必要的属性。单击“删除”按钮，可以删除当前选中的属性。单击“完成”按钮，将创建 JavaMail 资源。

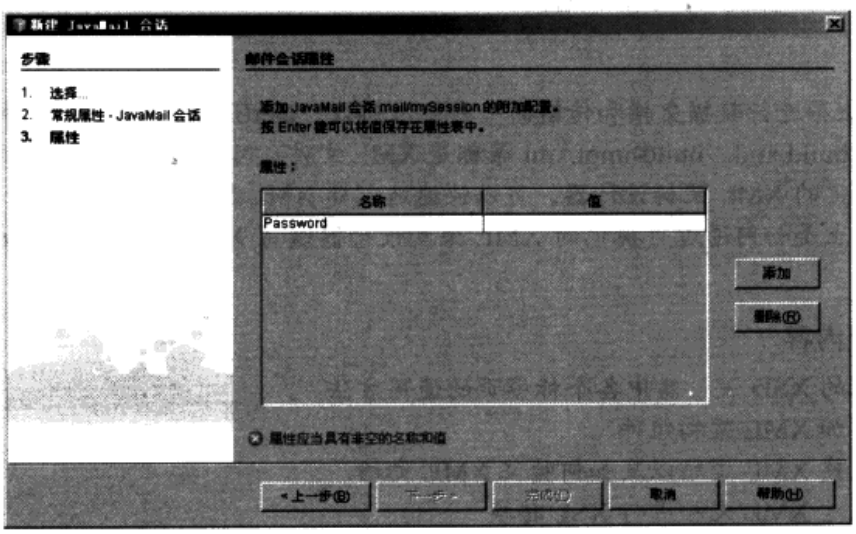


图 15-35 “邮件会话属性”对话框

15.5 小结

本章主要介绍企业级应用程序的开发方法。企业应用程序重要包括商业逻辑实现部分和客户端部分。其中，商业逻辑部分可以使用 EJB 模块或者持久性 API 来实现，是本章的介绍重点。

对于客户端的开发，例如 Web 应用程序的开发，因为前面几章已经进行了介绍，因此并不是本章的重点学习内容。读者在阅读的时候，重点了解如何与 EJB 模块进行交互即可。通过本章的学习，读者应该掌握以下基本内容：

- 开发一个企业应用程序的基本流程
- 实体 Bean、会话 Bean 和消息驱动 Bean 的开发方法
- 如何在客户端发送消息
- 如何使用持久性 API
- 如何创建几种 Sun 资源

Download at http://www.pin5i.com/

第 16 章

基于 XML 的开发

XML 是业界进行数据交换和传输的一种标准协议。NetBeans 的项目配置和编译文件，例如 build.xml、build.xml、build-impl.xml 等都是 XML 文件。NetBeans 提供了对 XML 开发的支持，通过可视化的 XML 架构设计器，可以快速地创建 XML 架构。

本章的重点是如何通过可视化的 XML 架构设计器编写 XML 架构，以及如何开发和编辑 XML 文档。

本章主要内容

- 可视化的 XSD 设计器中各个标签页的使用方法
- 如何添加 XML 架构组件
- 如何创建 XML 文档以及如何验证 XML 文档
- 如何通过 XML 文档进行 XSL 转换
- 如何使用 CSS 编辑器

本章亮点

- 查找 XML 组件架构组件的使用实例
- 如何通过 DTD 文档生成 DOM 和 SAX 处理代码
- 如何根据 XML 生成 DTD 文档

16.1 NetBeans 对 XML 开发的支持

在 NetBeans 5.x 中已经提供了 XML 的开发支持。Enterprise Pack 5.5 开发包提供了可视化的 XML 编辑环境，通过向导可以创建和修改 XML 框架，查看架构元素之间的关系，并且可以使用查询来查找全局组件的使用实例和未使用的全局组件。此外，Enterprise Pack 5.5 还提供了更多的 XML 开发支持。图 16-1 是未安装 Enterprise Pack 5.5 开发包时的 XML 节点，即 NetBeans 集成开发环境本身所能创建的 XML 文件类型。

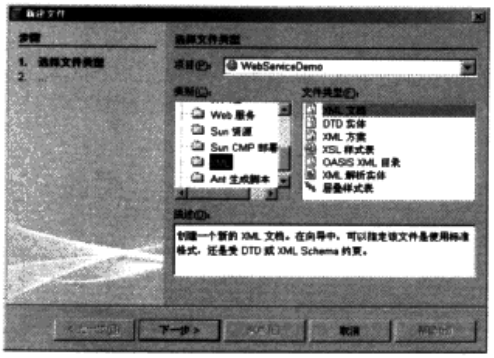


图 16-1　未安装 Enterprise Pack 5.5 开发包时的 XML 节点

Download at http://www.pin5i.com/

图 16-2 是安装了 Enterprise Pack 5.5 开发包后的 XML 节点，可以看到多了对一些新文件类型的支持。另外，提供的“样例构架”节点中还包括了“订单构架”和“贷款申请构架”两种典型的应用案例。

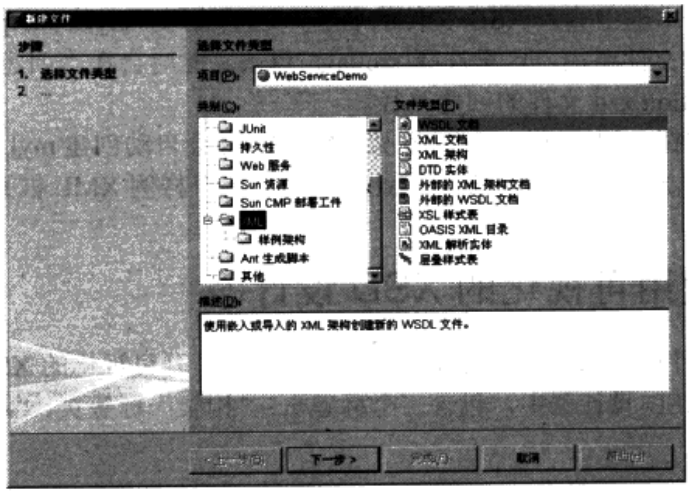


图 16-2 安装 Enterprise Pack 5.5 开发包后的 XML 节点

在 NetBeans 6.0 中，如果安装了包含全部功能的安装包，那么就默认支持以上所有的 XML 开发。这种改进简化了不少的操作和步骤，初学者无须详细地弄清楚每个开发包支持哪些功能，因此是一个很好的消息。下面将主要介绍如何在 NetBeans 6.0 中创建 XML 架构、编辑 DTD 实体、编辑 XML 文档等内容。

16.2 创建一个样例 XML 架构

本节将使用向导，创建一个订单构架样例 XML 框架，然后介绍 XML 开发的集成开发环境。

新建一个名字为 XMLSampleApp 的 Web 应用程序，并设置为主项目。在 NetBeans 集成开发环境中，选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”列表中选择“XML”节点，在右边的“文件类型”列表中选择“贷款申请架构”节点。单击“下一步”按钮，打开“新建 贷款申请样例架构”对话框，如图 16-3 所示。

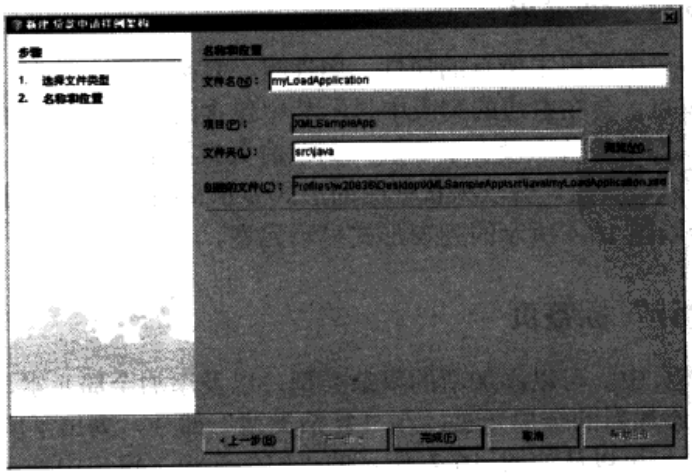


图 16-3 “新建 贷款申请样例架构”对话框

Download at http://www.pin5i.com/



说明：在“新建文件”对话框中，如果在“类别”列表中选择“XML”节点，然后在右边的“文件类型”列表中选择“XML 架构”节点，可以创建一个空白的 XML 构架。

在“文件名”文本框中输入 XSD 文件的名字，这里输入“myLoadApplication”。单击“浏览”按钮，选择保存文件路径，这里选择 src\java 目录。此时会在“创建的文件”文本框中，看到 myLoadApplication.xsd 文件的创建路径。

单击“完成”按钮，创建样例 XML 框架。NetBeans 会自动创建 myLoadApplication.xsd 文件，并打开可视化的 XSD 文件设计器。至此，完成了一个样例 XML 框架的建立。



16.3 使用可视化的 XSD 设计器

在创建 XSD 文件后，NetBeans 集成开发环境会自动打开可视化的 XSD 设计器，如图 16-4 所示。在可视化的 XSD 设计器中，包含三个标签页：“构架”标签页、“源”标签页和“设计”标签页。

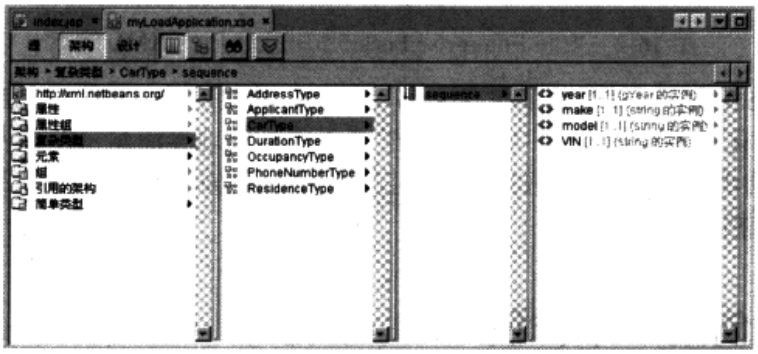


图 16-4 可视化的 XSD 设计器



说明：也可以在“文件”窗口中通过选择对应目录下的 XSD 文件，来打开 XSD 设计器。



16.3.1 “构架”标签页

在“构架”标签页中列出了所有的属性、属性组、复杂类型、元素、组、引用的框架和简单类型。单击某个节点，会在右边的列表中显示其中包含的内容。包含子节点的节点会使用旁边的黑色箭头表示。如果某节点没有子节点，则该箭头将显示为浅灰色。

单击工具栏上的 按钮，会以树状视图的形式显示文档元素，如图 16-5 所示。单击工具栏上的 按钮，会以图 16-4 所示的列表形式显示元素。



16.3.2 “设计”标签页

在“设计”标签页中，可以添加新的复杂类型，以及添加全局元素，如图 16-6 所示。在“元素”部分，可以添加新的全局元素；在“复杂类型”部分，列出了所有的复杂类型，以及类型包含的元素。用户可以从如图 16-7 所示的“组件面板”窗口中拖曳组件到“设计”标签页中添加新的复杂类型，或者定义新的全局变量。

Download at http://www.pin5i.com/

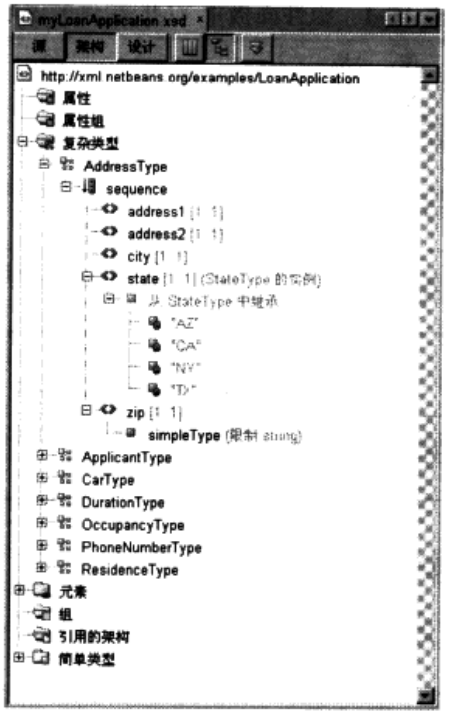


图 16-5　树状视图的形式显示文档元素

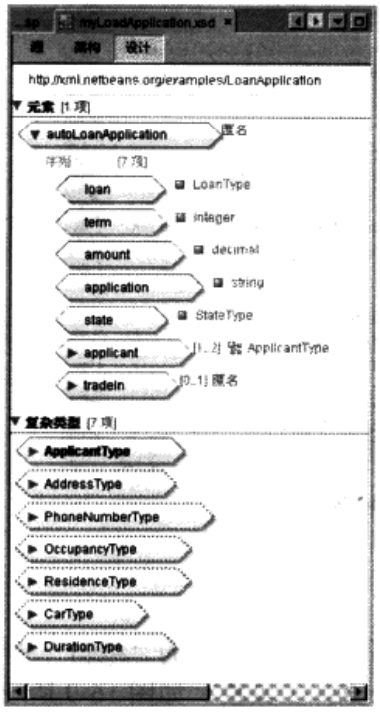


图 16-6　“设计”标签页

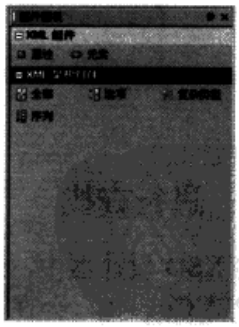


图 16-7　“组件面板”窗口

在组件面板中，包含以下两个部分的组件。

- XML 组件：可存在于 XML 文档或 XML 架构中的组件。
- XML 架构组件：只能存在于 XML 架构中的组件。

这些组件的说明如表 16-1 所示。

Download at http://www.pin5i.com/

表 16-1 组件面板中 XML 组件的说明

分 类	组件名称	图 标	说 明
XML 组件	属性	属性	属性组件用于定义架构属性
	元素	元素	元素组件用于定义架构元素
XML 架构组件	全部	全部	全部组件指定了子组件可以按任意顺序出现。此外，每个子组件可以不出现或出现一次
XML 架构组件	选项	选项	选项组件表示的是可选的元素，也就是说只能选择一个包含在<choice>声明中的元素
	复杂类型	复杂类型	复杂类型组件用于定义复杂类型。复杂类型组件是包含其他组件和/或属性的 XML 组件
	序列	序列	序列组件指定了子组件必须按顺序出现。每个子组件可以出现任意次数

16.3.3 “源”标签页

选择“源”标签页，可以直接浏览并修改 XML 文本格式的 myLoadApplication.xsd 文件，如图 16-8 所示。



图 16-8 “源”标签页

16.3.4 进行查找

可视化的 XSD 设计器中，选中“架构”标签页，选择菜单中的“编辑”→“查找”命令，或者按下快捷键 Ctrl + F，会在页面的下面打开一个查找工具栏，如图 16-9 所示。



图 16-9 查找工具栏

在文本框中输入要查找的内容，单击“查找下一个”按钮执行查找。如果被查找的内容有多个，单击“查找下一个”按钮继续查找下一个。单击 按钮，会打开一个弹出菜单。

如果选择弹出菜单中的“从选定内容中搜索”命令，则将从最近在“架构”视图中选择的组件开始查找操作。也就是说，将在选定的子树内执行搜索，而不是从架构文档的根目录中进

Download at http://www.pin5i.com/

行搜索。

如果选择弹出菜单中的“使用正则表达式”命令，则将使用文本字段中输入的正则表达式（区分大小写）来查找组件。

如果不希望查找组件，则选择以下的弹出菜单命令，搜索不同的内容。

- “组件名称”命令：查找名称属性与搜索短语相匹配的组件。
- “组件种类”命令：查找种类与搜索查询相匹配的组件。
- “属性声明”命令：查找其属性具有特定名称的组件。
- “属性值”命令：查找其属性具有特定值的组件。
- “名称空间使用实例”：查找引用特定名称空间的组件。

16.3.5 添加 XML 架构组件

可视化的 XSD 设计器的“架构”标签页和“设计”标签页，都可以添加 XML 架构组件。关于如何在“设计”标签页中添加组件，读者可以参考 SOA 程序开发实例中的介绍。本节主要介绍如何在“架构”标签页中添加 XML 架构组件。

在“元素”节点上单击鼠标右键，选择弹出菜单中的“添加元素”命令，打开“添加元素”对话框，如图 16-10 所示。

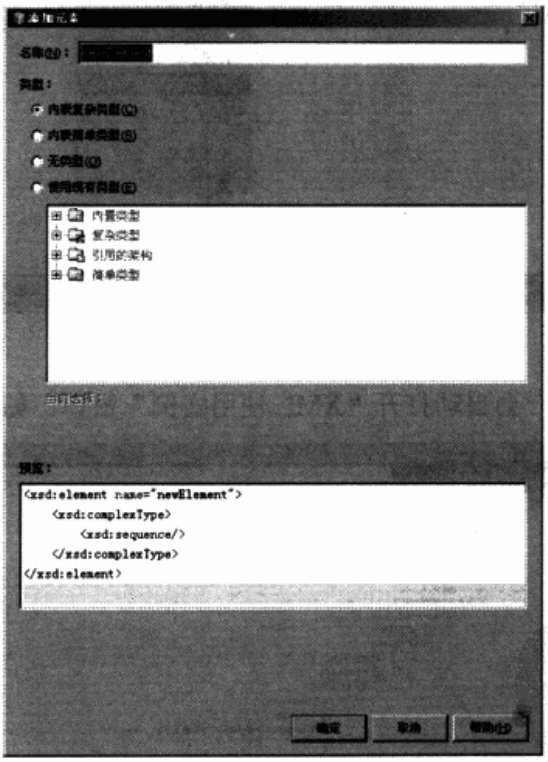


图 16-10 “添加元素”对话框

在“名称”文本框中输入元素的名字。元素的类型有如下几种。

- “内联复杂类型”单选按钮：将使用内联的复杂类型，自动添加<complexType> <sequence/> </complexType>标记。
- “内联简单类型”单选按钮：将使用内联的简单类型，自动添加<simpleType> <restriction base = "string"/> </simpleType>标记。
- “无类型”单选按钮：不使用任何的类型，即仅创建一个空元素。

Download at http://www.pin5i.com/

- “使用现有类型”单选按钮：使用现有的内置类型、复杂类型、引用的架构、简单类型等。当选中该单选按钮后，需要在下面的树状视图中选择要使用的现有类型。

在设置不同的类型时，可以通过下面的“预览”文本框查看要创建的 XML 标记。单击“完成”按钮，即可创建一个元素。



说明：添加其他组件（例如“属性”和“简单类型”的组件）的对话框和“添加元素”对话框类似，只是其中一些选项有所不同，这里不再赘述。



16.3.6 查找 XML 架构组件的使用实例

在可视化的 XSD 设计器的“架构”标签页中，右键单击某一个组件，然后选择弹出菜单中的“查找使用实例”命令，可以查找该组件的使用实例，如图 16-11 所示。

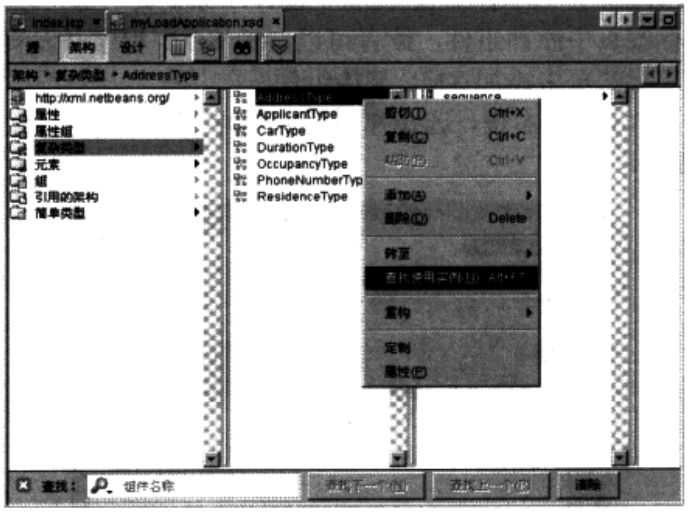


图 16-11 查找组件的使用实例

在查找到使用实例后，会自动打开“XML 使用实例”窗口，如图 16-12 所示。



图 16-12 “XML 使用实例”窗口

在树状窗格中，包含了查询结果的节点。双击某个节点，会自动在源代码编辑器中定位对应的代码行。单击左边工具栏上的“显示图形”按钮，显示如图 16-13 所示的布局窗口，显示各个节点之间的关系。

单击布局窗格中的节点，可以把节点拖曳到任意的位置。在布局窗格中单击鼠标右键，然后向上拖动可以缩小整个布局图形；在布局窗格中单击鼠标右键，然后向下拖动可以放大整个布局图形。

Download at http://www.pin5i.com/

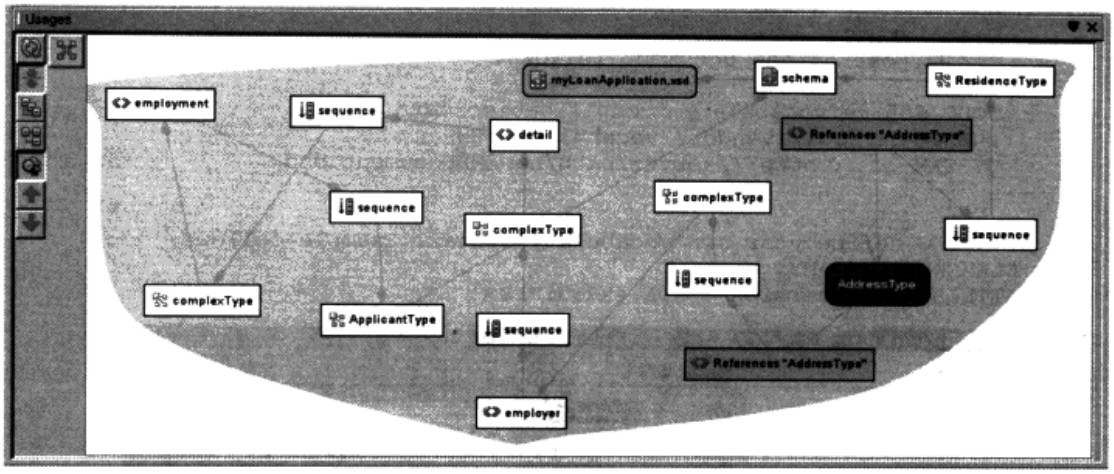


图 16-13 布局窗口

技巧：如果布局图形太大不能全部显示，可以缩小图形显示。另外，也可以单击“布局图”按钮，以显示整个图形。

在最左边的是工具栏，包含如下几个按钮。

- “刷新架构数据”按钮：当架构文件发生变化后，单击该按钮，可以刷新架构数据，使得节点和布局图反映最新的数据。
- “折叠树中的所有节点”按钮：把树状窗格中所有节点折叠。
- “显示逻辑视图”按钮：打开显示逻辑关系的视图。
- “显示物理视图”按钮：打开显示物理关系的视图。
- “上一个实例”按钮：定位到上一个实例。
- “下一个实例”按钮：定位到下一个实例。

16.4 创建和编辑 DTD 实体

在 NetBeans 中除了可以创建和编辑 XML 架构以外，还可以创建其他几种不同类型的 XML 文档，例如 DTD 实体、XSL 式样表等。本节主要介绍 DTD 实体的创建和操作方法。文档类型定义 DTD（Document Type Definition）用于描述 XML 文件中所使用的标记，并提供这些标记的有效布局。

16.4.1 通过向导创建 DTD 实体

在“新建文件”对话框中的“类别”列表中，选择 XML 节点，将在右边的“文件类型”列表中选择“DTD 实体”节点。单击“下一步”按钮，打开“新建 DTD 实体”对话框，如图 16-14 所示。

在“文件名”文本框中输入新建 DTD 文件的名字，在“文件夹”文本框中输入或者选择 DTD 文件的创建路径。单击“完成”按钮，自动创建如下的代码：

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
    Document   : newXMLDocument.dtd
    Created on : 2007 年 7 月 22 日, 下午 3:54
```

Download at http://www.pin5i.com/


```
    Author     : Robin
    Description:
        Purpose of the document follows.

    TODO define vocabulary identification data
    PUBLIC ID  : -//vendor//vocabulary//EN
    SYSTEM ID  : http://server/path/newXMLDocument.dtd
-->

<!-- TODO define your own vocabulary/syntax. Example follows:  -->
<!ELEMENT root ANY>
<!ATTLIST root version CDATA #REQUIRED>
```

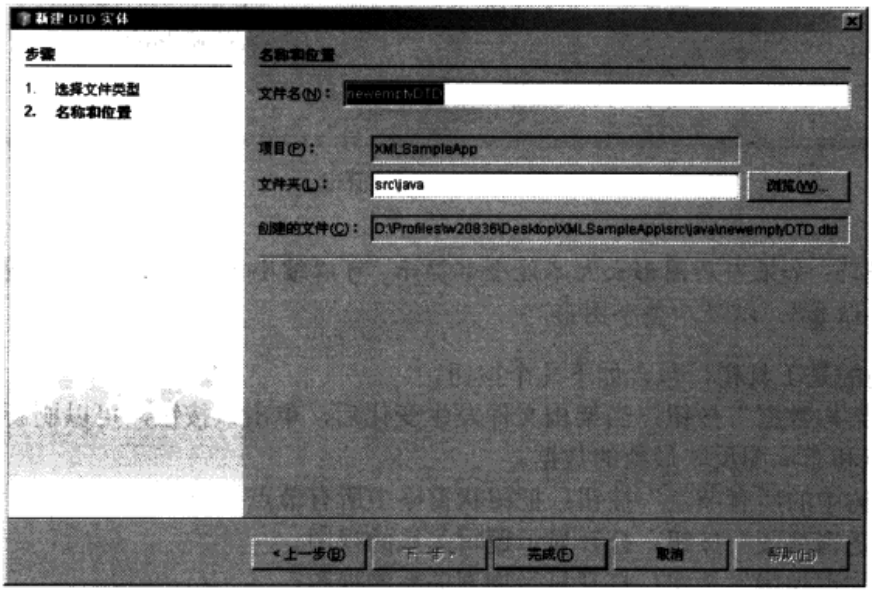


图 16-14 “新建 DTD 实体”对话框

16.4.2 检查 DTD 文档

在源代码编辑器中打开 DTD 文档。单击鼠标右键，选择弹出菜单中的“检查 DTD”命令，可以检查 DTD 文件的有效性。在输出窗口会输出检查的结果，例如在图 16-15 所示的检查结果中显示了一个错误。

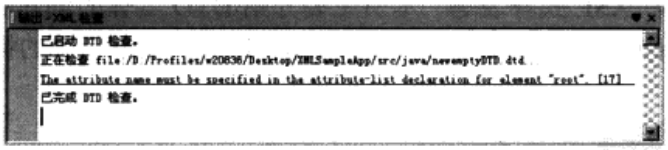


图 16-15 检查 DTD 文件的输出

16.4.3 生成 CSS 和文档

在源代码编辑器中打开 DTD 文档。单击鼠标右键，选择弹出菜单中的“生成 CSS”命令，可以为 DTD 文件创建一个式样表 CSS（关于式样表的编辑和修改请参考第 16.5 节）。

如果选择弹出菜单中的“生成文档”命令，会打开“选择文件名 *.html”对话框。在该对话框的“文件”文本框中输入文件的名字后，单击“确定”按钮会自动生成一个 HTML 文档。在浏览器中打开该文档，显示如图 16-16 所示的内容。

Download at http://www.pin5i.com/

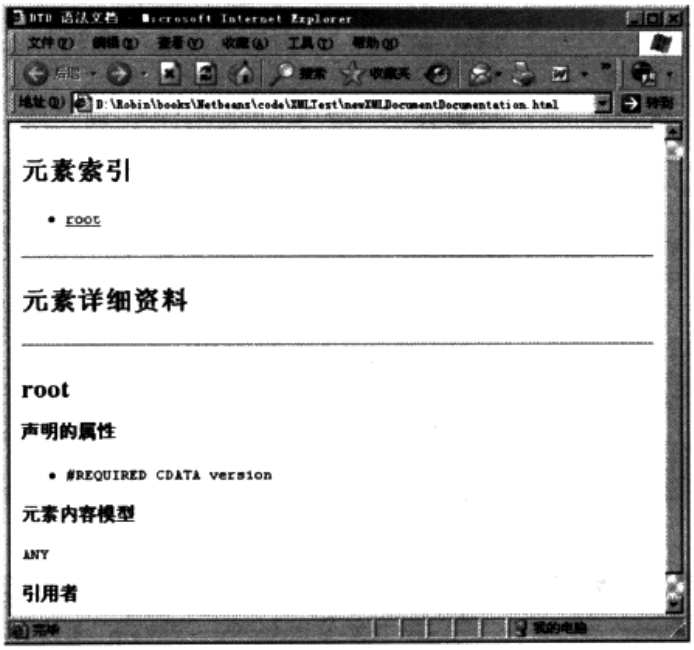


图 16-16 生成的 HTML 文档

16.5 创建和编辑 XML 文档

16.5.1 通过向导创建 XML 文档

在“新建文件”对话框中选择“XML 文档”节点后，单击“下一步”按钮，打开“新建 XML 文档”对话框，如图 16-17 所示。

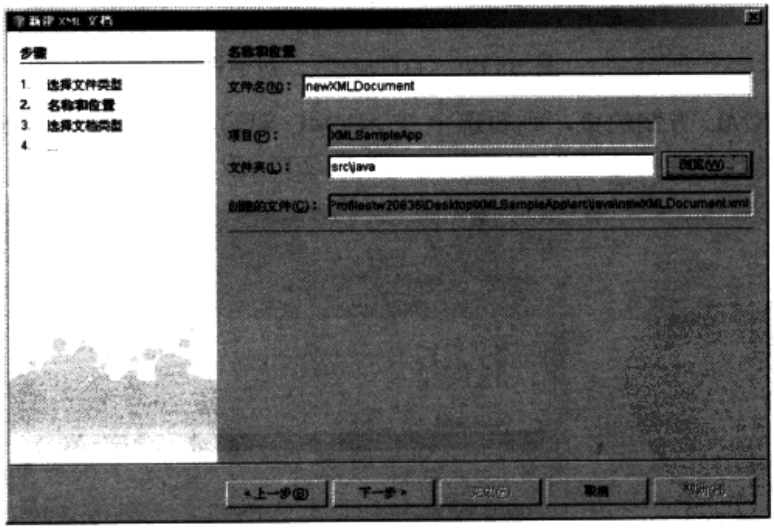


图 16-17 “新建 XML 文档”对话框

在“文件名”文本框中输入 XML 文档的名字。在“文件夹”文本框中输入或者选择 XML 文档的创建路径。单击“下一步”按钮，打开“选择文档类型”对话框，如图 16-18 所示。

如果希望添加 DTD 约束，则要选中“受 DTD 约束的文档”单选按钮，单击“下一步”按钮，打开“DTD 选项”对话框，如图 16-19 所示。

Download at http://www.pin5i.com/

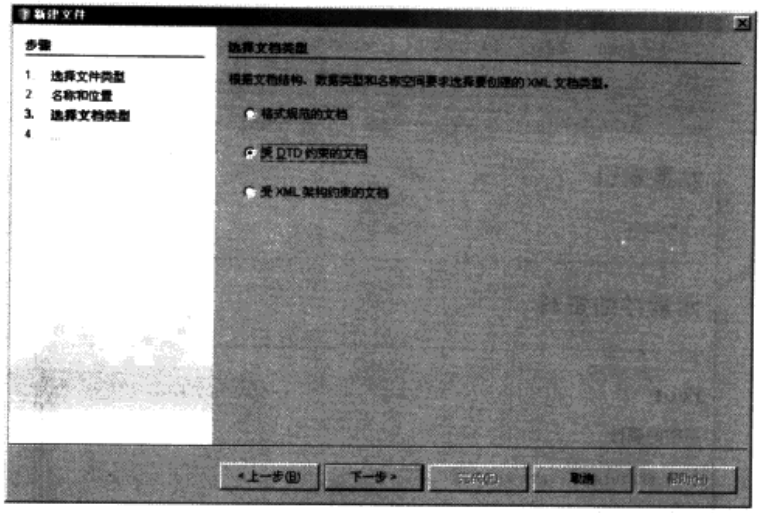


图 16-18 “选择文档类型”对话框

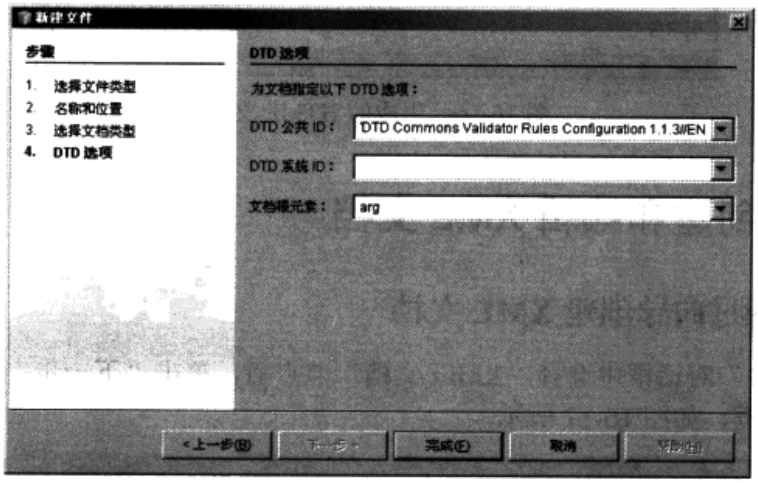


图 16-19 “DTD 选项”对话框

如果希望受 XML 方案约束，则要选中“受 XML 架构约束的文档”单选按钮，单击“下一步”按钮，会打开“架构选项”对话框，如图 16-20 所示。

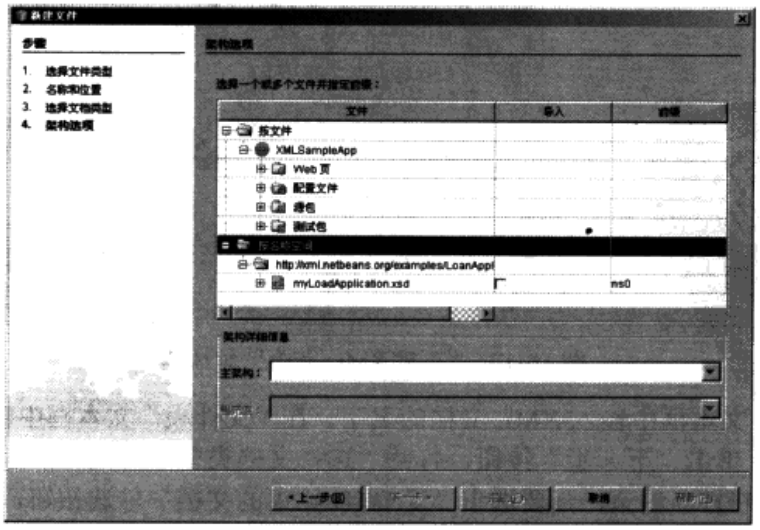


图 16-20 “架构选项”对话框

Download at http://www.pin5i.com/

如果选择“格式规范的文档”单选按钮，则可以直接单击“完成”按钮创建 XML 文档。创建的示例代码如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
    Document   : newXMLDocument.xml
    Created on : 2007年7月22日, 下午5:49
    Author     : Robin
    Description:
        Purpose of the document follows.
-->

<root>

</root>
```

16.5.2 查看 XML 文档

在源代码编辑器中打开 XML 文档。单击鼠标右键，选择弹出菜单中的“视图”命令，可以打开浏览器查看 XML 文档，如图 16-21 所示。

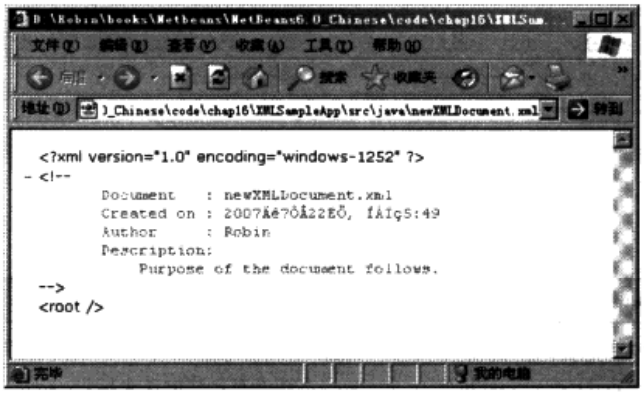


图 16-21 通过浏览器查看 XML 文档

16.5.3 检查 XML 文档

在源代码编辑器中打开 XML 文档。单击鼠标右键，选择弹出菜单中的“检查 XML”命令，或者按下快捷键 Alt + F9，将检查 XML 文档是否规范。检查的内容包括以下内容：

- 每个开始标记都必须具有结束标记；
- 元素不能重叠；
- 必须有且只有一个根元素；
- 必须将属性值用引号引起来；
- 元素不能具有两个同名的属性；
- 注释和处理指令不能出现在标记内；
- 未转义的<或&符号不能出现在元素的字符数据或属性的字符数据中。

检查的输出结果如图 16-22 所示。

16.5.4 验证 XML 文档

如果 XML 文档受 XML 架构或 DTD 约束，则可以验证这个文档来检查其内容中是否有语法错误。在源代码编辑器中打开 XML 文档，例如一个 Books.xml 文档：

Download at http://www.pin5i.com/

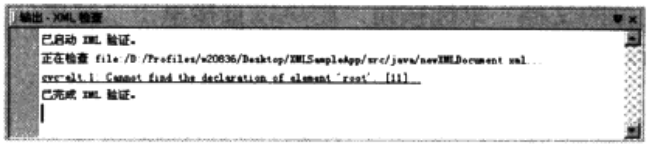


图 16-22 XML 检查的输出结果

```
<?xml version="1.0" encoding="gb2312" ?>
<books>
  <book>
    <name>精通 NetBeans 开发</name>
    <author>Robin</author>
  </book>
  <book>
    <name>精通 JavaME 程序开发</name>
    <author>RL</author>
  </book>
</books>
```

单击鼠标右键，选择弹出菜单中的“验证 XML”命令，或者按下快捷键 Alt + Shift + F9，将对 XML 进行验证。验证的输出结果如图 16-23 所示。在根据 DTD 文档验证时，生成的一些常见错误消息如表 16-2 所示。

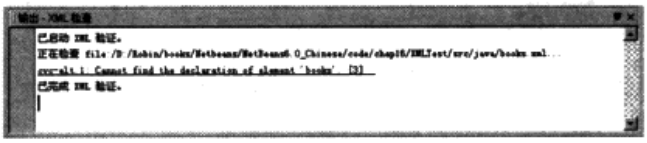


图 16-23 XML 验证的输出结果

表 16-2 XML 验证的一些常见错误消息

错误消息	含 义
Failed to read schema document	无法读取结构文档。请检查指定的 XML 方案是否可以被访问
End of entity not allowed; an end tag is missing	文档没有最终的结束标记
Expected "</tagname>" to terminate element started on line line#	指出的元素没有结束标记
Element type "element name" must be declared	XML 文档中包含未在 DTD 中声明的元素类型
Cannot find the declaration of element 'xxx'	未定义 DTD 文档，或者在 DTD 文档中没有定义 xxx 元素



注意：如果 XML 文档的 DTD 位于远程计算机上，请确保能够在 NetBeans 集成开发环境中访问该文件，例如应该正确地设置代理，并保证网络连接正常。如果不能访问指定的 DTD 文档，则会返回一个列出 DOCTYPE 声明的错误信息。



16.5.5 根据 XML 文档生成 DTD

在第 16.4.1 节中已经介绍了如何创建一个空 DTD 文档，本节将介绍通过现有的 XML 文档自动生成 DTD。

在源代码编辑器中打开上一节中的 Books.xml 文档。单击鼠标右键，选择弹出菜单中的“生成 DTD”命令，打开“选择文件名*.dtd”对话框。在“文件”文本框中输入要生成的 DTD 文件的名字，单击“确定”按钮创建 DTD 文档。下面是根据此文档自动生成的 DTD 文档：

```
<?xml version='1.0' encoding='UTF-8'?>
```

Download at http://www.pin5i.com/


```
<!--
    TODO 定义了词汇标识
    PUBLIC ID: -//vendor//vocabulary//EN
    SYSTEM ID: http://server/path/books.dtd

--><!--
    关于如何在 XML 文档中使用此 DTD 的示例:

    <?xml version="1.0"?>

    <!DOCTYPE books SYSTEM "books.dtd">

    <books>
    ...
    </books>
-->

<!--- 在此处放置 DTDDoc 注释。 -->
<!ELEMENT books (book)*>

<!--- 在此处放置 DTDDoc 注释。 -->
<!ELEMENT book (author|name)*>

<!--- 在此处放置 DTDDoc 注释。 -->
<!ELEMENT name (#PCDATA)>

<!--- 在此处放置 DTDDoc 注释。 -->
<!ELEMENT author (#PCDATA)>
```

可见，在自动生成的 DTD 文档中，还给出了如何使用该 DTD 文档的示例代码。下面把<!DOCTYPE books SYSTEM "books.dtd">添加到 Books.xml 文档中：

```
<?xml version="1.0" encoding="gb2312" ?>

<!DOCTYPE books SYSTEM "books.dtd">

<books>
…
</books>
```

然后再验证 XML，此时如图 16-23 所示的错误会消失，显示验证通过，如图 16-24 所示。

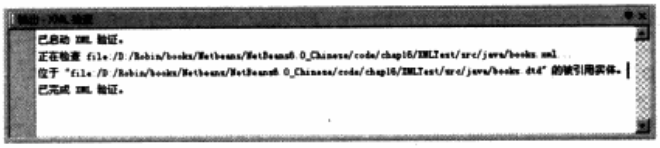


图 16-24 显示验证通过

16.6 创建可扩展样式表

可扩展样式表 XSL（Extensible Style Sheet）文档用于指定 XML 文档的显示属性、标记转换和寻址机制。使用 XSL 样式表可以转换任何 XML 文档中的数据，例如把 XML 文档中的数据转换为 HTML 或文本文件。

16.6.1 通过向导创建 XSL 式样表

在“新建文件”对话框中的“类别”列表中选择“XML”节点，将在右边的“文件类型”列表中选择“XSL 式样表”节点。单击“下一步”按钮，打开“新建 XSL 式样表”对话框，如图 16-25 所示。

Download at http://www.pin5i.com/

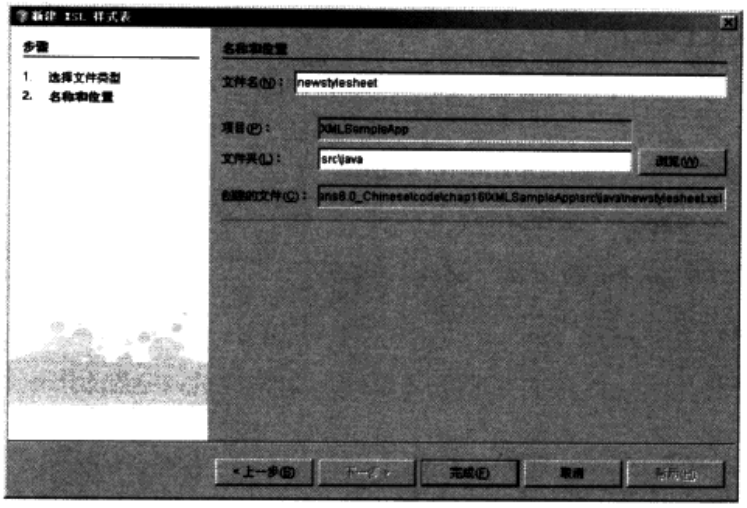


图 16-25 “新建 XSL 式样表”对话框

在“文件名”文本框中输入新建 XSL 式样表的名字，在“文件夹”文本框中输入或者选择 XSL 式样表文件的创建路径，这里输入“src\java”。单击“完成”按钮，创建一个式样表，代码如下所示：

```
<?xml version="1.0" encoding="UTF-8" ?>

<!--
    Document   : newstylesheet.xsl
    Created on : 2007年7月23日, 上午7:22
    Author     : Robin
    Description:
        Purpose of transformation follows.
-->

    <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
        <xsl:output method="html"/>

        <!-- TODO customize transformation rules
             syntax recommendation http://www.w3.org/TR/xslt
        -->
        <xsl:template match="/">
            <html>
                <head>
                    <title>newstylesheet.xsl</title>
                </head>
                <body>
                </body>
            </html>
        </xsl:template>

    </xsl:stylesheet>
```

修改 XSL 代码如下所示，用于在 HTML 文件中输出 Books.xml 文档中的全部内容。

```
        <xsl:template match="/">
            <html>
                <head>
                    <title>newstylesheet.xsl</title>
                </head>
                <body>
                    <xsl:text>在 HTML 中输出 Books.xml</xsl:text>
                    <hr></hr>
                    <p></p>
```

Download at http://www.pin5i.com/


```
                <xsl:element name="books">
                    <xsl:copy-of select="."/>
                </xsl:element>
            </body>
        </html>
    </xsl:template>
```

16.6.2 转换 XSL 式样表

在 XML 文档和 XSL 式样表中，都可以进行 XSL 转换。在源代码编辑器中，单击鼠标右键，选择弹出菜单中的“XSL 变换”命令，打开“XSL 变换”对话框，如图 16-26 所示。如果是通过 XSL 式样表打开的该对话框，则需要从“XML 源”下拉列表框中指定 XML 文档，或者单击旁边的“浏览”按钮进行查找。如果是从 XML 文档中打开该对话框，则需要从“XSLT 脚本”下拉列表框中选择要用于转换文档的 XSL 样式表，或者单击旁边的“浏览”按钮进行查找。

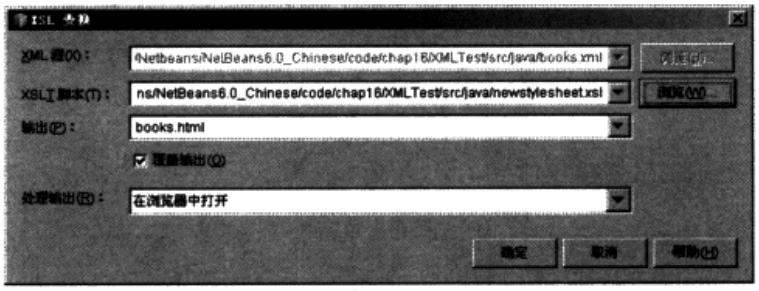


图 16-26 “XSL 变换”对话框

“输出”下拉列表框中指定转换的输出形式。默认情况下，NetBeans 生成与 XML 文档同名的 HTML 文件。如果 XSL 样式表创建一个不同类型的输出，会在“输出”组合框中自动指定此输出。

如果要更改输出路径的名称和位置，请在“输出”下拉列表框中输入相对路径和文件名。选中“覆盖输出”复选框，则输出文件已经存在时会自动覆盖。

在“处理输出”下拉列表框中可以指定处理输出文件的方式，可以有以下几种选项。

- “不执行任何操作”：只创建文件。
- “应用默认操作”：对输出文件应用 NetBeans 的默认操作。对大部分文件执行的默认操作是在源代码编辑器中打开文件。
- “在浏览器中打开”：将在默认的浏览器中打开输出文件。

单击“确定”按钮，执行 XSL 转换。在默认的浏览器中，打开输出的 Books.html 文档，显示界面如图 16-27 所示。

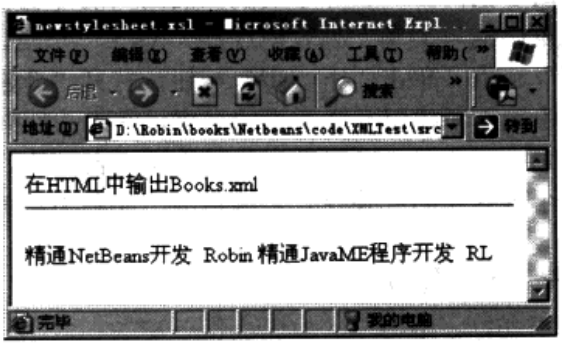


图 16-27 输出的 Books.html 文档显示界面

Download at http://www.pin5i.com/

16.7 创建 OASIS XML 目录文件和 XML 解析实体

OASIS XML 目录文件会将外部实体引用映射到本地资源，这样就可以在“运行环境”窗口中将目录添加到 DTD 和 XML Schema 目录中。

在“新建文件”对话框的“类别”列表中选择“XML”节点，在右边的“文件类型”列表中选择“OASIS XML 目录文件”节点。单击“下一步”按钮，打开“新建 OASIS XML 目录”对话框，如图 16-28 所示。

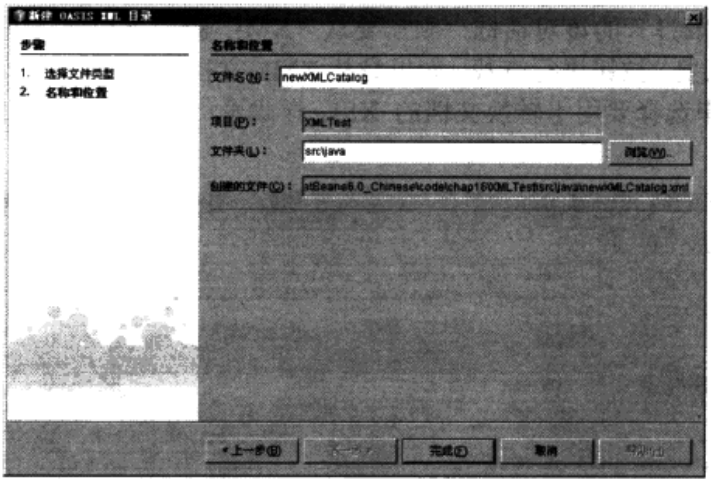


图 16-28 “新建 OASIS XML 目录”对话框

在“文件名”文本框中输入新建 OASIS XML 目录文件的名字，在“文件夹”文本框中输入或者选择 XSL 式样表文件的创建路径，这里输入 src\java。单击“完成”按钮，将创建一个 OASIS XML 目录文件。

说明： 创建 XML 解析实体的过程和创建 OASIS XML 目录文件的过程类似，这里不再赘述。

16.8 创建层叠样式表

层叠样式表 CSS（Cascading Style Sheet）文件主要用于对显示界面进行配置，可以设置布局、字体、颜色、背景等不同的内容。CSS 通常在网页设计中被使用，编写或者修改简单的代码，就可以改变同一页面的不同部分，甚至不同页数的网页的外观和格式。

在“新建文件”对话框中的“类别”列表中选择“XML”节点，将在右边的“文件类型”列表中选择“层叠样式表”节点。单击“下一步”按钮，打开“新建层叠样式表”对话框。

在“文件名”文本框中输入新建层叠样式表的名字，在“文件夹”文本框中输入或者选择该表的创建路径。单击“完成”按钮，将创建一个层叠样式表文件。在源代码编辑器中打开该文件后，会在下边显示一个可视化的层叠样式表编辑器，如图 16-29 所示。

16.8.1 层叠样式表编辑器的“字体”面板

层叠样式表编辑器的“字体”面板中，可以设置与字体相关的值。其中，“字体系列”用于设置要使用的字体，即设置层叠样式表的 font-family 属性。

Download at http://www.pin5i.com/

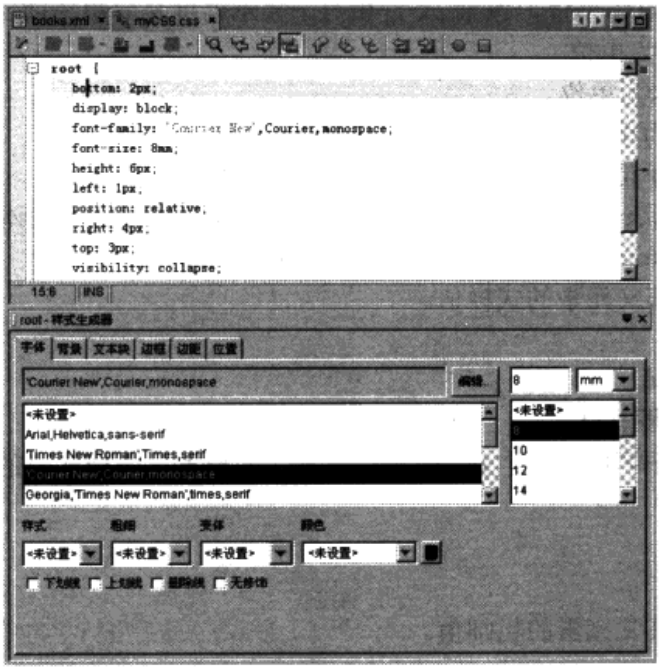


图 16-29 层叠样式表编辑器的“字体”面板

单击“编辑”按钮，将打开“编辑字体系列”对话框，如图 16-30 所示。在“当前字体系列”列表中列出了所有可以使用的字体系列，单击某个系列则会在“选定”列表中显示该系列中的字体。

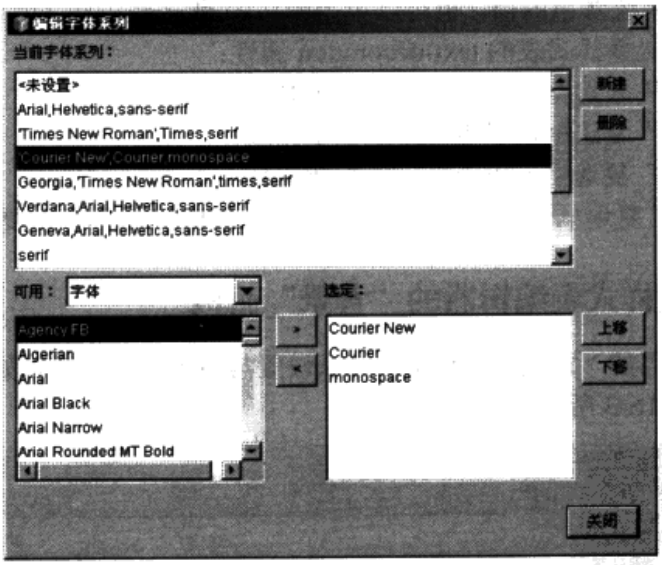


图 16-30 “编辑字体系列”对话框

选定后，单击“确定”按钮，返回层叠样式表编辑器的“字体”面板。窗口左边的文本框用于设置字体的大小，即设置层叠样式表的 font-size 属性。可以在文本框中直接输出字体大小，也可以在下面的列表框中选择大小。“大小”文本框旁边的下拉列表框用于设置字体大小的单位，可以使用的单位有以下几种。

- %：使用百分比。
- em：使用相对长度，即相对于当前对象内文本的字体尺寸。如当前对象内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸。

Download at http://www.pin5i.com/

- px：使用像素点为单位，像素 px 是相对于显示器屏幕分辨率而言的。
- cm：使用厘米为单位。
- mm：使用毫米为单位。
- in：使用英寸为单位。
- pt：使用磅为单位。

在“样式”下拉列表框中，用于设置以正常形式呈现字体还是以斜体或倾斜形式呈现字体，即设置层叠样式表的 font-style 属性。可以使用如下几个样式。

- inherit：继承其父元素的式样值。
- italic：倾斜。
- normal：正常体。
- oblique：倾斜罗马体。

“粗细”下拉列表框用于设置文字的粗细程度，即设置层叠样式表的 font-weight 属性。可以使用如下几个值。

- bold：黑体。
- bolder：较黑。
- inherit：继承其父元素的粗细值。
- lighter：较轻。
- normal：正常。

“变体”下拉列表框用于设置是否以小写形式呈现字体，即设置层叠样式表的 font-variant 属性。

“颜色”下拉列表框用于设置字体的颜色，即设置层叠样式表的 font-color 属性。下拉列表框中给出的值，都是系统颜色的值。也可以单击旁边的 按钮，打开颜色选择器进行选择。在修饰部分，可以设置层叠样式表的 text-decoration 属性：

- 选中“下画线”复选框可以为文字添加下画线；
- 选中“上画线”复选框可以为文字添加上画线；
- 选中“删除线”复选框可以为文字添加删除线；
- 选中“无修饰”复选框，则不使用任何的修饰，同时上面的三个复选框将变为禁止状态。



16.8.2 层叠样式表编辑器的“背景”面板

层叠样式表编辑器的“背景”面板中，可以设置与背景相关的值，如图 16-31 所示。其中，各个选项的说明如表 16-3 所示。

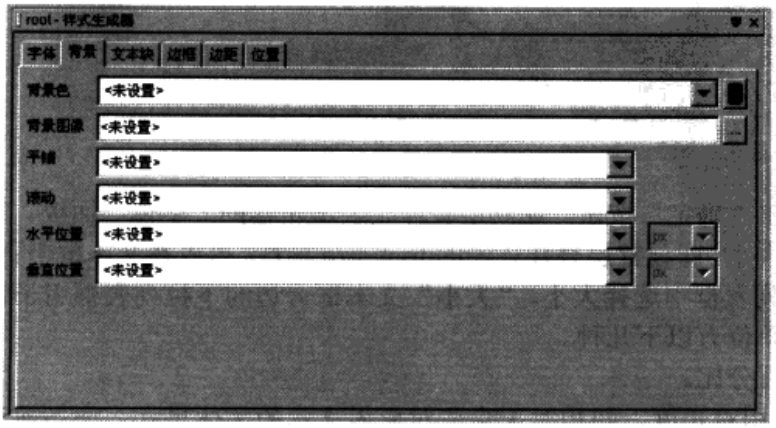


图 16-31 层叠样式表编辑器的“背景”面板

Download at http://www.pin5i.com/

表 16-3 层叠样式表编辑器的“背景”面板选项

选 项	含 义
“背景色”下拉列表框	用于设置要使用的背景颜色，即设置层叠样式表的 background-color 属性
“背景图像”下拉列表框	用于设置作为组件背景的图像，即设置层叠样式表的 background-image 属性。可以单击旁边的 ... 按钮查找图片
“平铺”下拉列表框	用于设置背景图像是出现一次，还是以不同的方式重复出现，即设置层叠样式表的 background-repeat 属性。可以选择如下几个值。 • inherit：继承其父元素的平铺值。 • no-repeat：图像不重复，但在其初始位置（左上角或使用“水平位置”和“垂直位置”设置的位置）上显示。 • repeat：图像平铺或重复，直至其覆盖整个背景。 • repeat-x：水平方向重复。图像在跨越背景宽度的行中重复，默认情况下跨越图像的顶部或在“垂直位置”设置的位置上。 • repeat-y：垂直方向重复。图像在跨越背景高度的列中重复，默认情况下位于图像的左侧或在“水平位置”设置的位置上
“滚动”下拉列表框	设置层叠样式表的 background-attachmen 属性。这个属性和背景图像属性一起使用，决定图片是跟随内容滚动，还是固定不动。可以有如下两个选项。 • fix：使图像固定，并且在内容移动时保持位置不变。也就是说，背景图像未连接到内容上，在用户滚动内容时，图像不随之移动。例如，可以将页面背景的水印设置为固定，以使水印在页面滚动时可见。 • scroll：使图像随组件的内容一起滚动（默认设置值）。也就是说，背景图像连接到内容上，在用户滚动内容时，图像随之移动
“水平位置”下拉列表框	用于指定图像的水平初始位置，即设置层叠样式表的 background-position 属性
“垂直位置”下拉列表框	用于指定图像的垂直初始位置，即设置层叠样式表的 background-position 属性

图 16-31 中的设置，对应的 CSS 代码如下所示：

```
root {
   background-attachment: fixed;
   background-color: aqua;
   background-position: center right;
   background-repeat: repeat;
   display: block;
   font-family: 'Courier New',Courier,monospace;
   font-size: 8mm
}
```

16.8.3 层叠样式表编辑器的“文本块”面板

层叠样式表编辑器的“背景”面板中，可以设置与背景相关的值，如图 16-32 所示。其中，各个选项的说明如表 16-4 所示。

表 16-4 层叠样式表编辑器的“文本块”面板选项

选 项	含 义
“水平对齐”下拉列表框	用于设置文本在组件的左边距和右边距之内的水平对齐，即设置层叠样式表的 text-align 属性
“纵向排列”下拉列表框	用于设置组件内容相对于文本的基线或相对于同一行中其他对象的基线出现的位置，即设置层叠样式表的 vertical-align 属性

Download at http://www.pin5i.com/

续表

选　项	含　义
“缩排”下拉列表框	用于设置文本块内段落第一行的缩进量，即设置层叠样式表的 text-indent 属性
“方向”下拉列表框	用于设置读取文本的方向，即设置层叠样式表的 direction 属性。可以使用如下选项。 • inherit：继承其父元素的值。 • ltr：从左到右读取，例如英语、简体中文等。 • rtl：从右到左读取，例如希伯来语
“字间距”下拉列表框	该属性控制单词之间的间距，即设置层叠样式表的 word-spacing 属性
“字母间距”下拉列表框	该属性控制字母之间的间距，即设置层叠样式表的 letter-spacing 属性
“行高”下拉列表框	设置层叠样式表的 line-height 属性。该属性控制文本行之间的垂直间距。具体来说，行高是相邻文本行基线之间的距离

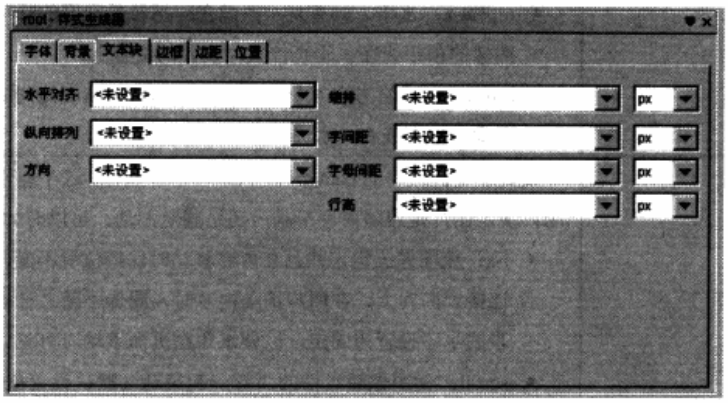


图 16-32　层叠样式表编辑器的“文本块”面板

图 16-32 中的设置，对应的 CSS 代码如下所示：

```
root {
   direction: ltr;
   display: block;
   font-family: 'Courier New',Courier,monospace;
   font-size: 8mm;
   letter-spacing: 2px;
   line-height: normal;
   text-align: center;
   text-indent: 1px;
   vertical-align: baseline;
   word-spacing: 1px
}
```



16.8.4　层叠样式表编辑器的“边框”面板

层叠样式表编辑器的“边框”面板中，可以设置与边框相关的值，如图 16-33 所示。其中，各个选项的说明如表 16-5 所示。

表 16-5　层叠样式表编辑器的“边框”面板选项

选　项	含　义
“样式”下拉列表框	用于设置绘制的边框类型，例如实线、虚线、点画线等
“宽度”下拉列表框	用于设置绘制的线条宽度
“颜色”下拉列表框	用于设置绘制线条的颜色

Download at http://www.pin5i.com/

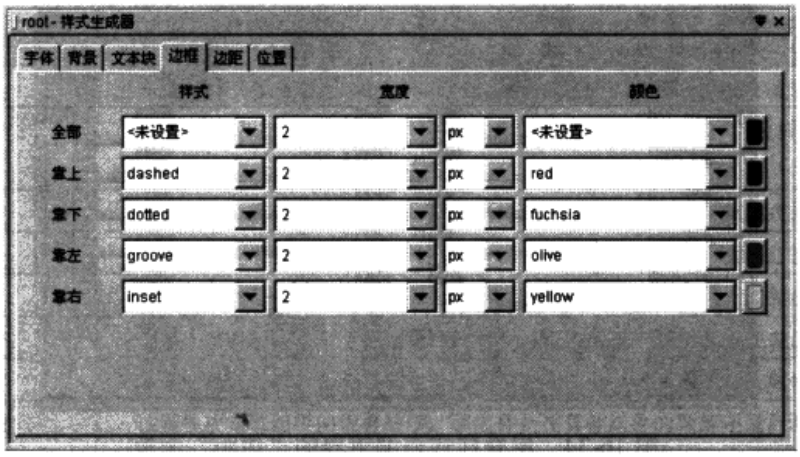


图 16-33 层叠样式表编辑器的“边框”面板

如果在“全部”下拉列表框中选择式样，那么该式样将应用于上、下、左、右四个边框，即设置 border 属性。

如果设置“靠上”的值，那么只影响上边框，相当于设置了 border-top 属性；如果设置“靠下”的值，那么只影响下边框，相当于设置了 border-bottom 属性；如果设置“靠左”的值，那么只影响左边框，相当于设置了 border-left 属性；如果设置“靠右”的值，那么只影响右边框，相当于设置了 border-right 属性。

图 16-33 中的设置，对应的 CSS 代码如下所示：

```
root {
   border-bottom-color: fuchsia;
   border-bottom-style: dotted;
   border-bottom-width: 2px;
   border-left-color: olive;
   border-left-style: groove;
   border-left-width: 2px;
   border-right-color: yellow;
   border-right-style: inset;
   border-right-width: 2px;
   border-top-color: red;
   border-top-style: dashed;
   border-top-width: 2px;
   display: block;
   font-family: 'Courier New',Courier,monospace;
   font-size: 8mm
}
```



16.8.5 层叠样式表编辑器的“边距”面板

层叠样式表编辑器的“边距”面板中，可以设置与边距相关的值，如图 16-34 所示。其中，各个选项的说明如表 16-6 所示。

表 16-6 层叠样式表编辑器的“边距”面板选项

选 项	含 义
“边距”下拉列表框	设置组件边界与任何相邻组件边界之间的间距。创建的间距在组件边界的外部
“填充”下拉列表框	设置边界框边缘与组件内容之间在组件边界框内的间距。创建的间距在组件边界的内部

Download at http://www.pin5i.com/

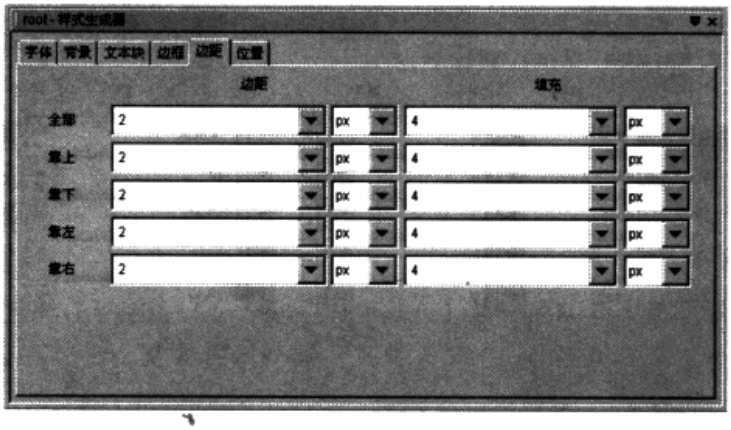


图 16-34 层叠样式表编辑器的“边距”面板

图 16-34 中的设置，对应的 CSS 代码如下所示：

```
root {
   display: block;
   font-family: 'Courier New',Courier,monospace;
   font-size: 8mm;
   margin-bottom: 2px;
   margin-left: 2px;
   margin-right: 2px;
   margin-top: 2px;
   padding-bottom: 4px;
   padding-left: 4px;
   padding-right: 4px;
   padding-top: 4px
}
```

16.8.6 层叠样式表编辑器的“位置”面板

层叠样式表编辑器的“位置”面板中，可以设置与位置相关的值，如图 16-35 所示。其中，各个选项的说明如表 16-7 所示。

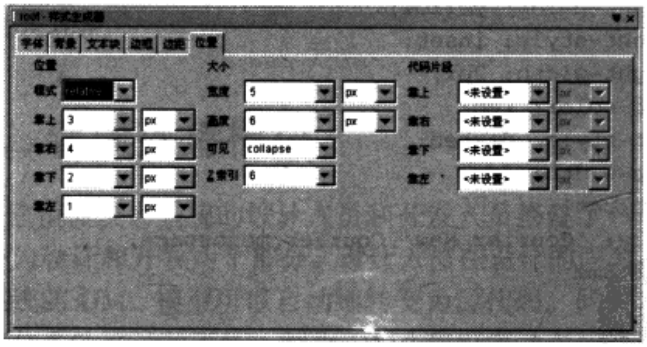


图 16-35 层叠样式表编辑器的“位置”面板

表 16-7 层叠样式表编辑器的“位置”面板选项

选 项	含 义
“模式”下拉列表框	设置位置的模式，决定着组件在页面上显示的位置。可以有如下几个选项。 • absolute：绝对定位。 • fixed：固定定位。 • inherit：继承父元素属性值。 • relative：相对定位。 • static：静态定位

Download at http://www.pin5i.com/

续表

选　项	意　义
“靠上”下拉列表框	该属性决定着要定位的组件与所包含块顶部之间的距离。层叠样式表的对应属性为 top
“靠下”下拉列表框	该属性决定着要定位的组件与所包含块底部之间的距离。层叠样式表的对应属性为 bottom
“靠左”下拉列表框	该属性决定着要定位的组件与所包含块左侧之间的距离。层叠样式表的对应属性为 left
“靠右”下拉列表框	该属性决定着要定位的组件与所包含块右侧之间的距离。层叠样式表的对应属性为 right
“宽度”下拉列表框	该属性决定着在将位置设置为“绝对”时组件的宽度。层叠样式表的对应属性为 width
“高度”下拉列表框	该属性决定着在将位置设置为“绝对”时组件的高度。层叠样式表的对应属性为 height
“可见”下拉列表框	该属性决定着组件是否显示在呈现的页面上。层叠样式表的对应属性为 visible
“Z 索引”下拉列表框	该属性决定着组件的堆叠顺序，即组件是在其他组件的顶部还是后面显示。使用负数值会将组件放在所包含块的后面。层叠样式表的对应属性为 z-index
“代码片段”下拉列表框	设置代码片段矩形，可以使组件边框与代码片段矩形边缘之间的组件部分变得透明，从而对所呈现组件部分有效地进行代码分段或剪切。层叠样式表的对应属性为 clip:rect



注意：如果将位置模式设置为静态，则“靠上”下拉列表框、“靠下”下拉列表框、“靠左”下拉列表框、“靠右”下拉列表框的设置不适用。

图 16-35 中的设置，对应的 CSS 代码如下所示：

```
root {
   bottom: 2px;
   display: block;
   font-family: 'Courier New',Courier,monospace;
   font-size: 8mm;
   height: 6px;
   left: 1px;
   position: relative;
   right: 4px;
   top: 3px;
   visibility: collapse;
   width: 5px;
   z-index: 6
}
```



16.9 小结

XML 已经被广泛使用在各种应用程序中，例如 Sun Java 应用程序服务器的配置文件、Web 服务的一些配置文档等。本章主要介绍基于 XML 相关的一些开发，例如创建 XML 架构、编写 XSL 式样等。通过本章的学习，读者应该掌握以下基本内容：

- 如何创建 XML 及其相关文档，包括 XML、DTD、XSL、OASIS XML、XML 解析实体、CSS 等；
- 如何对 XML 文档进行验证；
- 如何根据 XML 文档创建 DTD 文档；
- 如何转换 XSL 式样表；
- 如何使用可视化的 CSS 编辑器。

Download at http://www.pin5i.com/

第 17 章

UML 建模

统一建模语言 UML（Unified Modeling Language）是为团队开发提供的标准通用的设计语言，也就是像建筑行业使用的建筑图纸一样。目前，UML 已经获得了业界的广泛认同，越来越多的系统都使用 UML 来进行设计和开发。

NetBeans 提供了对 UML 的全面支持，利用 NetBeans UML 建模模块，可以创建 8 种类型的 UML 图：活动图、类图、协作图、组件图、部署图、序列图、状态图和用例图，并可以进行正向工程和逆向工程。本章将主要介绍如何在 NetBeans 中进行 UML 建模。

对于不需要学习 UML 建模的读者，可以越过本章继续向后学习。本章相对独立，跳过本章不会对后续的学习产生影响。在学习本章前，建议读者首先学习 UML 的基本知识。本章着重介绍 NetBeans 的使用方法，而不再对 UML 的一些基础知识进行阐述。

本章主要内容

- 如何下载和安装 UML 模块
- NetBeans 工作空间
- 源代码编辑器
- NetBeans 的模块安装/更新方法

本章亮点

- NetBeans 的安装
- NetBeans 模块的自动和手工安装方法
- NetBeans 集成开发环境



17.1 UML 建模支持模块

NetBeans 的 UML 建模模块为系统的设计人员和开发人员搭建了一座沟通的桥梁，从而使得开发过程不再明显分为设计和开发两个部分。设计人员在设计的同时，也参加了部分的程序开发工作，因为描述系统的 UML 模型可以自动地转变为源代码。同时，开发人员在编写代码的过程中，也参与了部分系统设计的工作，因为 UML 建模模块支持通过对源代码的更改来更新 UML 模型。



17.1.1 UML 建模模块的功能

利用 NetBeans UML 建模功能，设计人员和开发者可以分别专注于各自的工作，即设计人员专注于应用程序的设计，而开发人员则专注于编写代码。在设计和分析人员使用 UML 模型来设计系统后，开发人员就根据生成的 UML 模型来编写源代码。

NetBeans 的 UML 建模模块具有如下一些功能。

- 支持 8 种类型的 UML 图：活动图、类图、协作图、组件图、部署图、序列图、状态图

Download at http://www.pin5i.com/

和用例图。

- 生成无标记的代码：通过 UML 模型生成的代码不包含任何人工标记，例如注释或标记等。
- 收集设计需求：只需使用一个 XML 文件，就可以将应用程序的设计需求添加到模型中。
- 预定义且可扩展的设计模式：UML 模块提供了一套完整的设计模式，其中包括 EJB 1.1、EJB 2.0 等共 23 种模式。
- 生成 Web 报告：可以为自己的模型创建 Javadoc 形式的 Web 报告，从而与其他开发者轻松地共享所做的工作。
- 正向工程：通过模型来自动生成 Java 代码。
- 逆向工程：通过现有的 Java 源代码来创建 UML 模型。



17.1.2 下载和安装 UML 建模模块

在 NetBeans 5.x 集成开发环境中，UML 建模模块需要单独按钮。方法是选择“工具”→“更新中心”命令，打开更新中心向导。单击“下一步”按钮，打开“选择要安装的模块”对话框。在“可用更新和新模块”列表中，选择“UML 建模”选项，如图 17-1 所示。

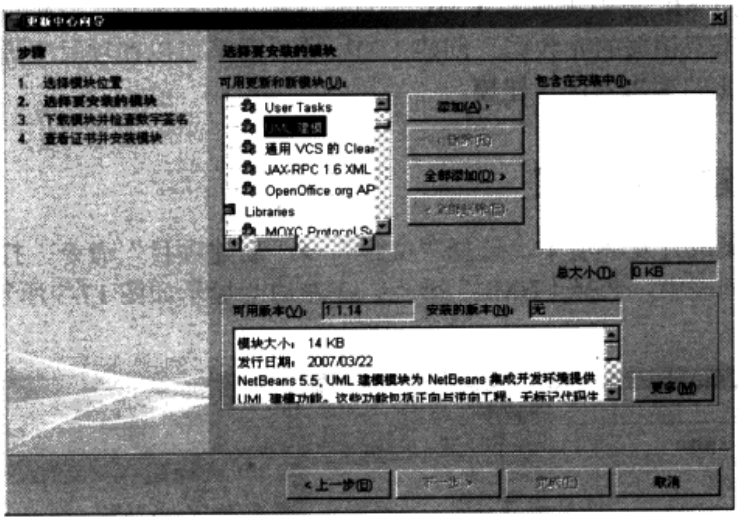


图 17-1 “选择要安装的模块”对话框

单击“添加”按钮，把 UML 模块添加到右侧的“包含在安装中”列表中，这时会在该列表中显示 UML 建模模块及其所需的所有相关库模块。单击“下一步”按钮，按照提示完成插件模块的安装。安装后将重新启动 NetBeans 5.x 集成开发环境，即可进行 UML 的建模设计了。

在 NetBeans 6.0 中简化了 UML 建模模块的安装，在如图 17-2 所示的下载界面中，只有选择下载包含了全部功能的 NetBeans，才能进行 UML 建模设计。安装完毕 NetBeans 6.0 集成开发环境后，就可以直接进行 UML 的开发了。



说明：UML 模块需要比较多的内存，建议为 NetBeans 至少设置 512MB 的使用内存。在 NetBeans 安装目录的 etc 子目录中，打开 netbeans.conf 文件。该文件是 NetBeans 的配置文件。在 netbeans_default_options 行中，如果有“-J-Xmx128m”，则表示 NetBeans 使用内存为 128MB，这里可以修改为“-J-Xmx512m”，使用 512MB 内存。

Download at http://www.pin5i.com/

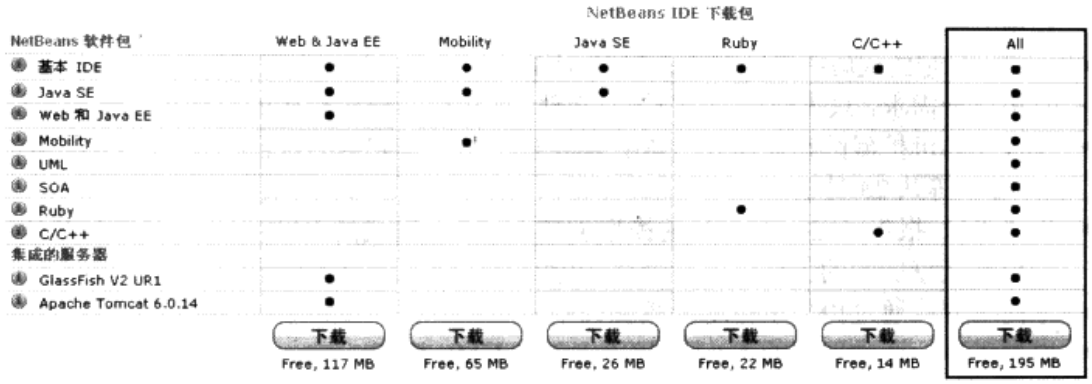


图 17-2 NetBeans 6.0 下载界面

17.2 创建 UML 项目以及 UML 建模环境

UML 建模使用 UML 项目来组织系统建模过程。在安装了 UML 建模模块后，就可以创建 UML 项目（也称为 UML 建模项目）。UML 项目包含模型的所有图、它们的关联元素、任何源文件以及与 UML 模型相关的元数据。创建 UML 项目后，可以在项目中通过模板创建这种图形，并可通过可视化的编辑器进行建模。

17.2.1 创建 UML 项目

在 NetBeans 集成开发环境中，选择“文件”→“新建项目”命令，打开“新建项目”对话框。在“类别”列表中，选择“UML”节点，在左边会显示如图 17-3 所示的几个 UML 项目类型。

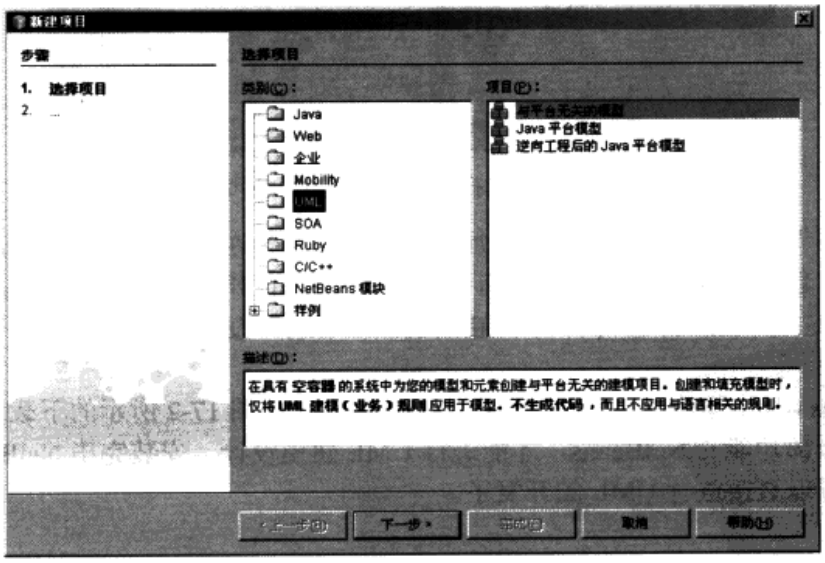


图 17-3 可以创建的 UML 项目类型

“与平台无关的模型”是指只有 UML 建模（与平台或者语言无关）规则会应用到项目创建的模型和元素中。创建和填充模型时，仅将 UML 建模（业务）规则应用于模型。不生成源代码，同时也不应用与平台或者语言相关的规则。

“Java 平台模型”是指创建与 Java 平台相关的 UML 项目，Java 平台或者语言规则会应用

Download at http://www.pin5i.com/

到 UML 项目创建的模型和元素中。如果目标开发语言是 Java，那么可以创建这样的一个 UML 模型项目。

“逆向工程后的 Java 平台模型”是指通过对现有 Java 项目执行逆向工程所获得的元素进行 UML 建模。

对 UML 逆向工程的详细介绍将在后面几节中进行。这里选择“Java 平台模型”节点，单击“下一步”按钮，打开“新建 Java 平台模型”对话框。

在“项目位置”文本框中输入项目所在的目录。在“项目名称”文本框中输入项目的名字，这里输入“UMLProjectDemo”。单击“完成”按钮创建一个 UML 项目。此时，会打开一个“创建新图”对话框，如图 17-4 所示。

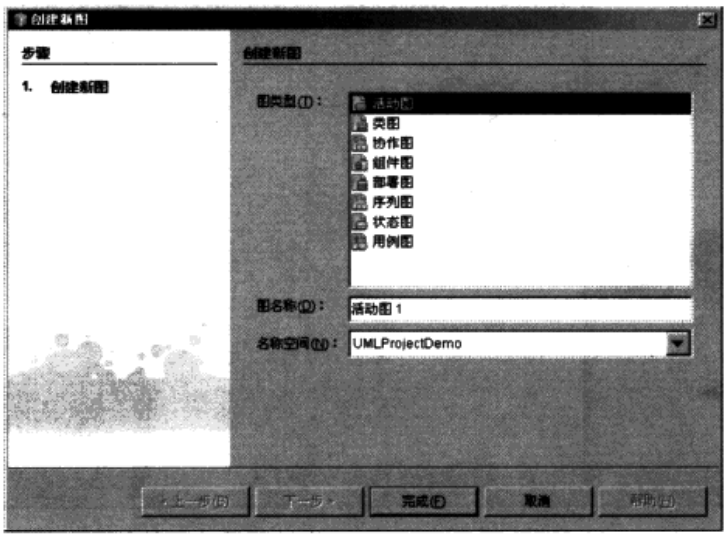


图 17-4 “创建新图”对话框

图是 UML 建模显示系统中静态结构的方法。图中包含两类数据：布局信息和表示法信息。其中，布局信息是在图编辑器中构造图的布局数据。例如，所有对象的水平和垂直位置、链接长度以及所有图组件的位置。NetBeans 在扩展名为*.etld 的二进制文件中保存此信息。

表示法信息是在图编辑器中构造图外观的装饰数据。例如：图中所有类元素的颜色、用于表示类属性的字体等。NetBeans 在扩展名为*.etlp 的可编辑 XML 文件中保存此信息。

在 UML 项目中，可以创建如下八种类型的图。

- 活动图：它是任何系统活动以及活动之间数据流或判断的可视表示。
- 类图：可视化地描述了系统设计中的类和接口以及它们之间的关系。该图可用于定义内部的、面向对象的代码结构。
- 协作图：强调发送和接收消息的对象结构组织的交互图。此类型的图显示围绕实例以及彼此间的链接组织的交互。
- 组件图：它是组件定义的可视表示，描述了系统的可部署单元以及一些接口的交互。该图可用于研究系统的体系结构。
- 部署图：它是节点及其上面部署的工件定义的可视表示。此类型的图基于使用部署元素扩展的常规静态结构图。
- 序列图：它是系统中协作对象组之间交互的可视表示，描述了某个功能需求的路径或相对时间的详细行为。该图可用于理解系统元素之间的消息流程。
- 状态图：描述了系统如何改变状态以响应内部的和外部的事件。该图可用于确保每个事件都被适当地处理，不管系统当前处于什么状态。

Download at http://www.pin5i.com/

- 用例图：从用户角度描述系统功能，并指出各功能的操作者。用例图描述了系统外的人员和系统的动作以及系统的响应。该类型的图提供系统的静态用例视图，可用于描述系统的功能需求。

如果创建一个空白的 UML 项目，因此单击“取消”按钮即可。这里，在“图类型”列表中选择“用例图”节点，在下面的“图名称”文本框中输入用例图的名字“Banker”。单击“完成”按钮，完成 UML 项目的创建。此时 NetBeans 集成开发环境的显示界面如图 17-5 所示。

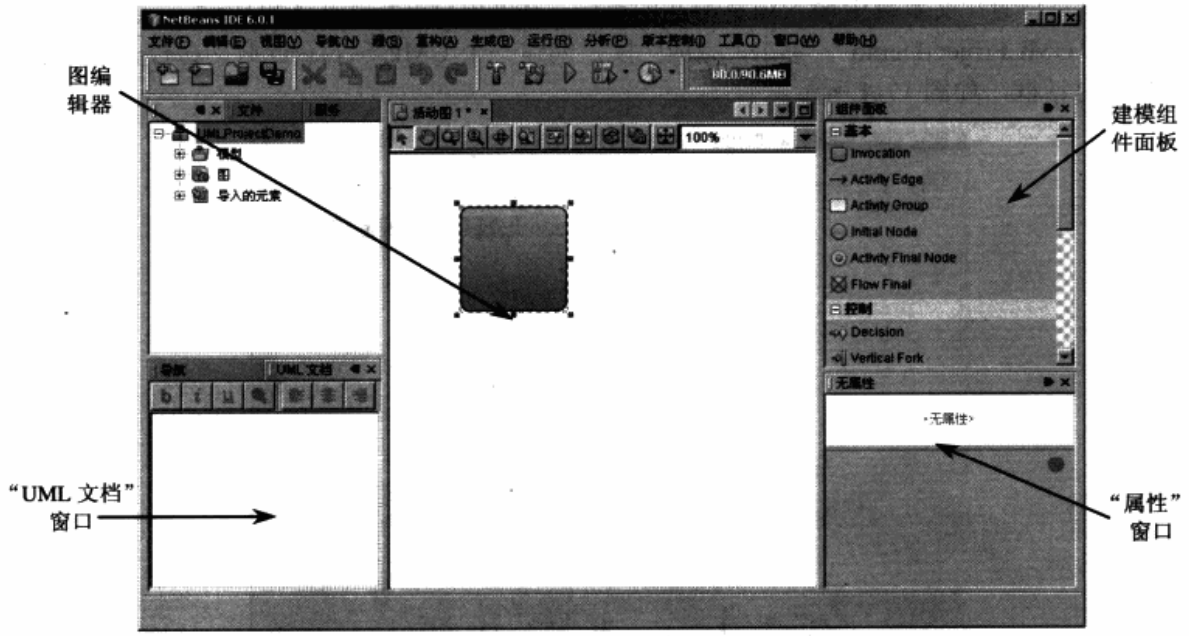


图 17-5 UML 设计环境

17.2.2 “文档”窗口

“文档”窗口如图 17-6 所示，用于显示有关任何 UML 模型元素的描述，并将该描述保存到相应项目的*.etd 文件中。当在图编辑器中选择元素或在“项目”窗口中选择节点时，“文档”窗口将显示对该元素的描述。

用户也可以在“文档”窗口的空白处直接编辑或者输入元素的描述信息。“文档”窗口的工具栏中提供了几个编辑文档用的按钮。



图 17-6 “文档”窗口

- “粗体”按钮 b：加粗突出显示的文本。对于突出显示的粗体文本，再次单击该图标可取消粗体。
- “斜体”按钮 i：倾斜突出显示的文本。对于突出显示的斜体文本，再次单击该图标可取消斜体。
- “下画线”按钮 u：为突出显示的文本加下画线。对于突出显示的带下画线的文本，再次单击该图标可取消下画线。
- “字体颜色”按钮：单击打开该按钮将打开“字体颜色选择器”对话框，如图 17-7 所示，从中可以为“文档”窗口中的字体选择新的颜色。
- “左对齐”按钮：用于将所有选定文本在“文档”窗口中左对齐。
- “居中”按钮：用于将所有选定文本在“文档”窗口中居中对齐。
- “右对齐”按钮：用于将所有选定文本在“文档”窗口中右对齐。

Download at http://www.pin5i.com/

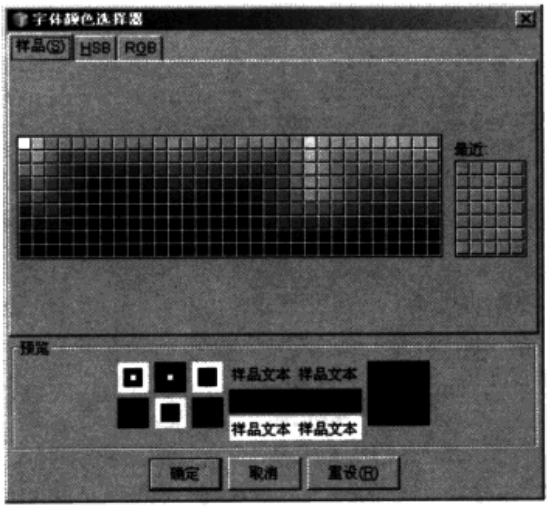


图 17-7 “颜色选择器”对话框

17.2.3 “属性”窗口

“属性”窗口如图 17-8 所示，用于显示有关在图编辑器或“项目”窗口中选择的任何建模元素属性的详细信息。

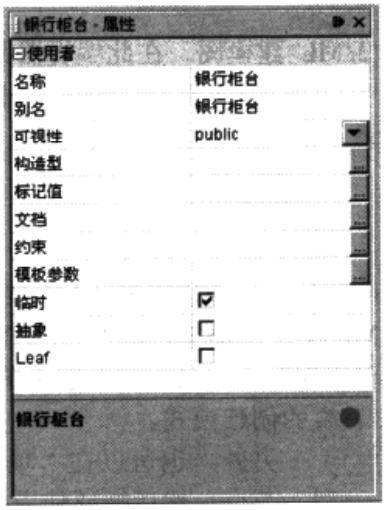


图 17-8 “属性”窗口

修改 UML 模型元素属性的方法和修改 SWING 组件属性的方法类似，这里不再赘述。在 UML 图编辑器中，右键单击要编辑属性的元素，然后选择弹出菜单中的“属性”命令，也自动打开一个“属性”窗口，用于编辑选中元素的属性。

17.2.4 UML 建模组件面板

在图编辑器中打开某个图时，建模组件面板中也会显示该图需要的组件。例如，打开类图时，组件面板中会显示如图 17-9 所示的类图元素。组件面板中各个元素的具体含义，都可以在 UML 建模标准文档中查到。这里只着重介绍 NetBeans 的功能，因此不再赘述，请读者自行参考必要的文档。

Download at http://www.pin5i.com/

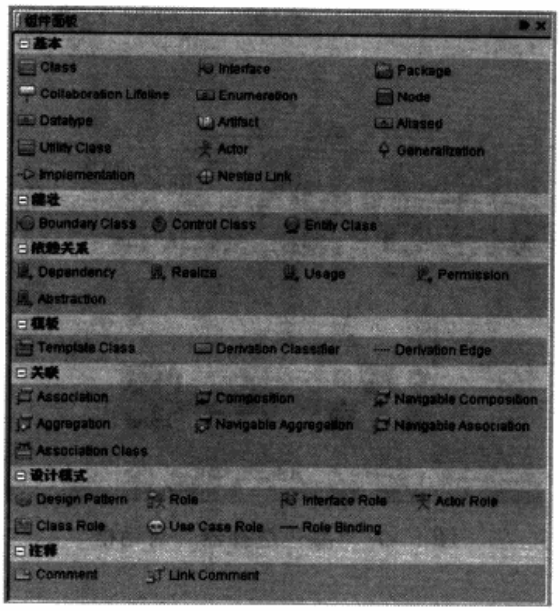


图 17-9 类图元素

17.3 UML 图编辑器

图编辑器用于显示或者编辑 UML 建模图。在此编辑器中，可以创建、编辑和删除 UML 模型的图元素。UML 图编辑器是 UML 建模的主要设计环境，因此本节将详细讨论如何在 UML 图编辑器中修改或者操作建模元素。

17.3.1 添加图元素

在创建了某个 UML 图后，可以从组件面板中向 UML 图中添加元素。单击组件面板要添加的组件，然后把鼠标移动到 UML 图编辑器中，此时鼠标会变为如图 17-10 所示的形状。单击鼠标，会添加一个元素。如果要添加多个元素，可以继续单击鼠标。右键单击以取消选择元素的创建模式。

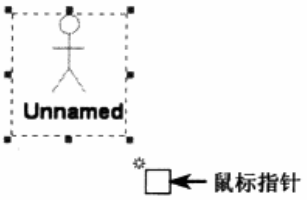


图 17-10 使用鼠标添加 UML 元素

另外，也可以在“项目”窗口中，展开项目下的“模型”节点，然后把要使用的模型拖曳到当前图中使用。

UML 图编辑器还支持从其他图中添加元素。首先在 UML 源图中，选择要重用的元素。然后从 NetBeans 的主菜单中选择“编辑”→“复制”命令，或者右键单击该元素，选择弹出菜单中的“编辑”→“复制”命令。最后，在目标图中单击鼠标右键，选择弹出菜单中的“编辑”→“粘贴”命令。这样，该元素将出现在目标图中，其标记将显示它的源图和原始名称。

17.3.2 复制、剪切和粘贴元素

在 UML 图编辑器中的元素，可以通过上下文菜单来进行编辑，如图 17-11 所示。其中，“复制”、“剪切”和“粘贴”命令用于复制、剪切和粘贴元素，这和 Word 中的文本操作类似。下面介绍其他几个编辑命令。

Download at http://www.pin5i.com/

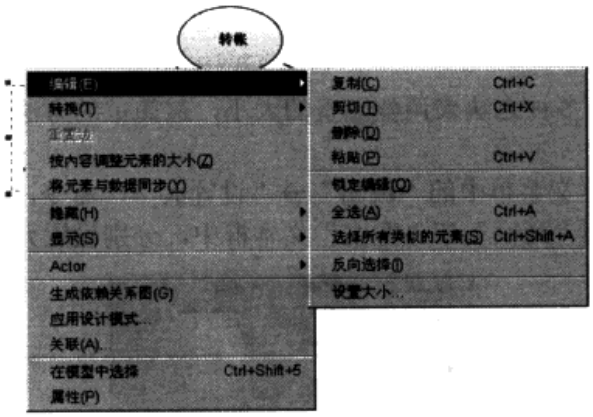


图 17-11 编辑 UML 图元素的命令

17.3.3 删除元素

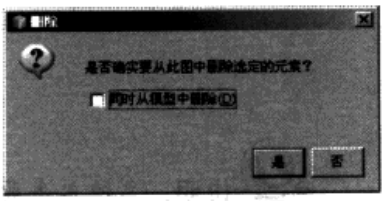


图 17-12 “删除”对话框

当选择“编辑”→“删除”命令时，会显示如图 17-12 所示的“删除”对话框。如果单击“是”命令，将只删除当前图中的元素。此时，在“项目”窗口中，打开项目下的“模型”节点，然后可以看到该元素的模型。如要彻底删除该元素，还需要删除模型节点下的该元素。

另外，一个彻底删除元素的简单方法，就是在如图 17-12 所示的对话框中选中“同时从模型中删除”复选框，然后再单击“是”按钮。

17.3.4 锁定编辑

到选择“编辑”→“锁定编辑”命令时，可以禁用图编辑器中元素属性和操作的编辑。被锁定的元素上会显示一个锁的标记，如图 17-13 所示。如果要解除锁定，只需要再次选择“编辑”→“锁定编辑”命令即可。



图 17-13 锁定的图元素

17.3.5 选择所有类似的元素

当选择“编辑”→“选择所有类似的元素”命令时，可以选中当前图中所有相同类型的元素。图 17-14 中，左边的图为没有执行“选择所有类似的元素”命令前的 UML，右边的图是执行该命令以后的图。

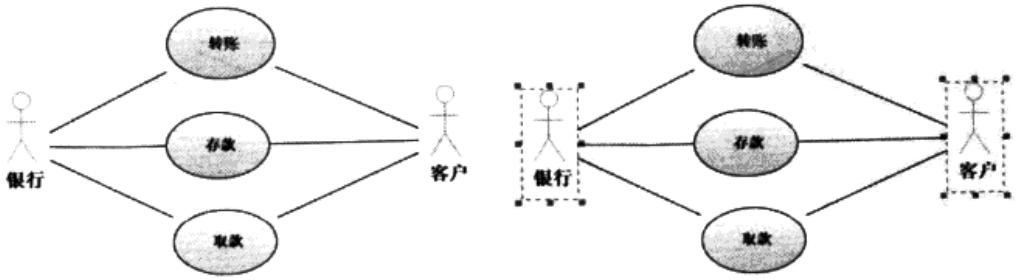


图 17-14 选择所有类似的元素

Download at http://www.pin5i.com/

17.3.6 调整元素大小

UML 图编辑器支持多种方法来调整元素的大小，最简单的方法是通过鼠标拖曳选中元素周围的小矩形。

另外，选择元素上下文菜单中的“编辑”→“设置大小”命令，会打开“设置大小”对话框，如图 17-15 所示。在“高度”和“宽度”文本框中，分别设置元素的高和宽即可。

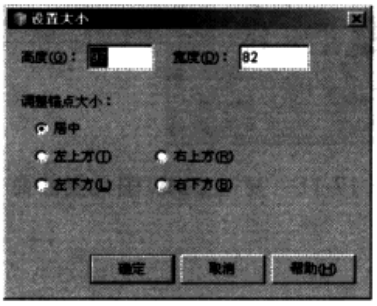


图 17-15 “设置大小”对话框

说明：右键单击要调整大小的组件，然后在弹出菜单中选择“按内容调整元素的大小”命令。UML 图编辑器会按一定的大小比例（能够清晰看到所有数据）来调整元素的大小。

17.3.7 堆叠图元素

UML 图编辑器还提供了如下的按钮，实现不同的堆叠功能。

- “前移”按钮：将元素后移一层（快捷键为 Ctrl+Alt+Shift+U）。
- “移至顶层”按钮：将元素移至底层（快捷键为 Ctrl+Alt+Shift+F）。
- “后移”按钮：将元素前移一层（快捷键为 Ctrl+Alt+Shift+D）。
- “移至底层”按钮：将元素移至顶层（快捷键为 Ctrl+Alt+Shift+B）。

17.3.8 删除所有拐点

在编辑 UML 图的时候，拖曳元素后连线经常会出现一些拐点，如图 17-16 所示。

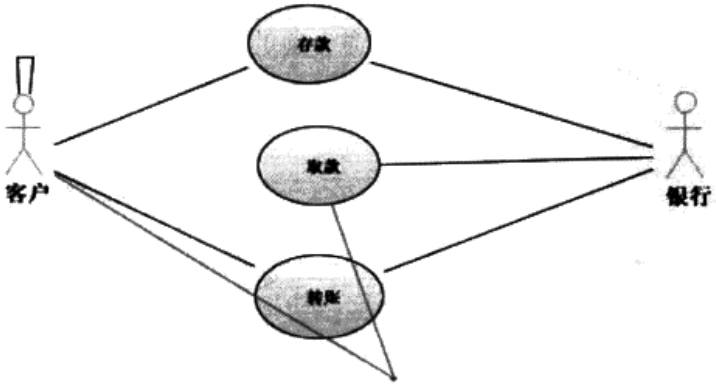


图 17-16 连线出现拐点

Download at http://www.pin5i.com/

此时，可以单击出现拐点的连线，选择弹出菜单中的“删除所有拐点”命令，这样就可以把折线变为直线了，如图 17-17 所示。

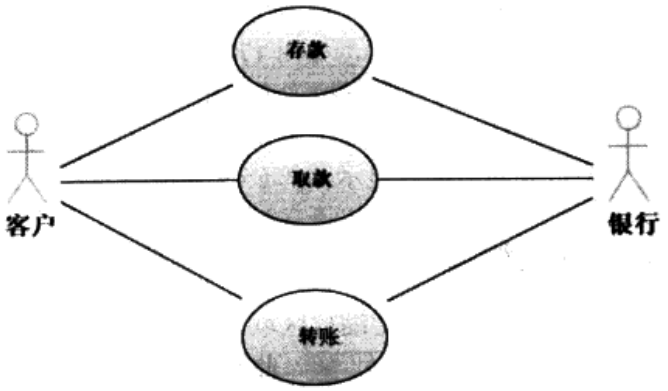


图 17-17　修复拐点后的 UML 图

17.4　UML 图形的缩放

UML 图编辑器可以以任意的比例显示 UML，即提供了如下几种缩放功能：

- 放大和缩小；
- 缩放到适当大小；
- 基于选择区域缩放；
- 交互缩放。

17.4.1　放大和缩小

单击 UML 图工具栏上的“放大”按钮，或者右键单击图中空白区域，选择弹出菜单中的“放大”命令，都可以使得 UML 图逐步放大；单击 UML 图工具栏上的“缩小”按钮，或者右键单击图中空白区域，选择弹出菜单中的“缩小”命令，都可以使得 UML 图逐步缩小。

在工具栏中，修改如图 17-18 所示的下拉列表框，也可以以任意的比例缩放图形。

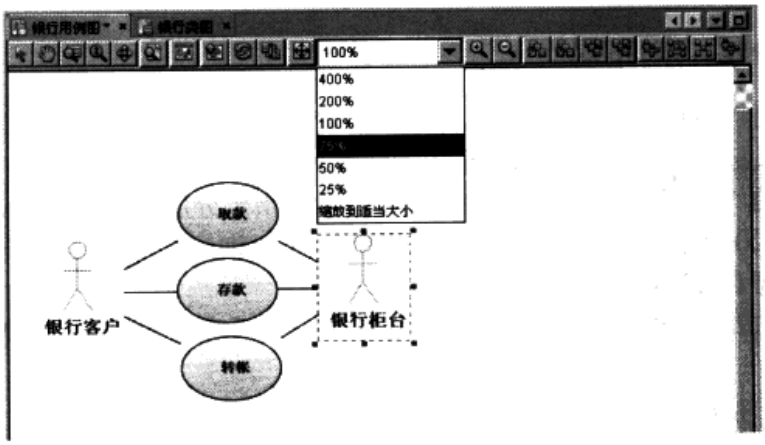


图 17-18　通过下拉列表框进行缩放

说明：放大功能的快捷键为“Ctrl++”；缩小功能的快捷键为“Ctrl+-”。

Download at http://www.pin5i.com/

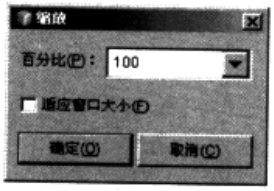


图 17-19 “缩放”对话框

17.4.2 缩放到适当大小

右键单击图中空白区域，选择弹出菜单中的“缩放”命令，会打开“缩放”对话框，如图 17-19 所示。

在“百分比”下拉列表框中，可以任意修改或者选择进行缩放的比例。如果选中“适应窗口大小”复选框，则 UML 图会在 UML 图编辑器的边框内显示图的全貌。

17.4.3 基于选择区域缩放

基于选择区域缩放功能，可以通过指定缩放区域来放大任意 UML 图形。在 UML 图编辑器的工具栏上单击“基于选择区域缩放”按钮，此时指针模式更改为“基于选择区域缩放”。在 UML 图编辑器中拖动一个矩形区域，该区域内的内容将被放大。

说明：基于选择区域缩放的快捷键为 Ctrl+Alt+Shift+Z。

17.4.4 交互缩放

交互缩放可以在 UML 图中将指针沿 *Y* 轴拖动来缩放任意缩放。在 UML 图编辑器的工具栏上单击“交互缩放”按钮，此时鼠标指针变为“交互缩放”模式。要对图进行放大，请将指针向下拖动；要对图进行缩小，请将指针向上拖动。

说明：交互缩放的快捷键为 Ctrl+Alt+Shift+I。

17.5 图布局

在 UML 图编辑器中，预设了几个图的布局。通过选取预设的布局，可以重新排列任意 UML 图的现有布局。

17.5.1 分层结构布局

分层结构布局将图重新排列为分层结构模式，使大多数元素位于引导元素的下面。在 UML 图的空白处单击鼠标右键，选择弹出菜单中的“布局”→“分层结构”命令，或者单击 UML 图编辑器工具栏的“分层结构布局”按钮，都可以指定分层布局。如果是第一次使用，还可能出现如图 17-20 所示的“布局”对话框。选中“不再询问”复选框，单击“是”按钮，这样下次就不会出现该对话框了。

对于如图 17-17 所示的 UML 图，使用分层结构布局后变为如图 17-21 所示的结果。

说明：分层结构布局的快捷键为 Ctrl+Alt+Shift+H。

17.5.2 对称结构布局

对称结构布局将图重新排列为方块形状，大致为矩形模式。在 UML 图的空白处单击鼠标右键，选择弹出菜单中的“布局”→“对称结构”命令，或者单击 UML 图编辑器工具栏的“对

Download at http://www.pin5i.com/

称结构布局”按钮，都可以指定对称布局。对于如图 17-17 所示的 UML 图，使用对称结构布局后，显示变为如图 17-22 所示。

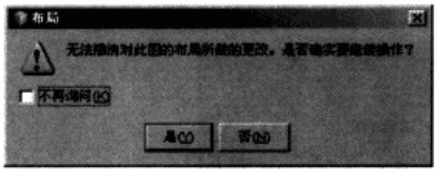


图 17-20 “布局”对话框

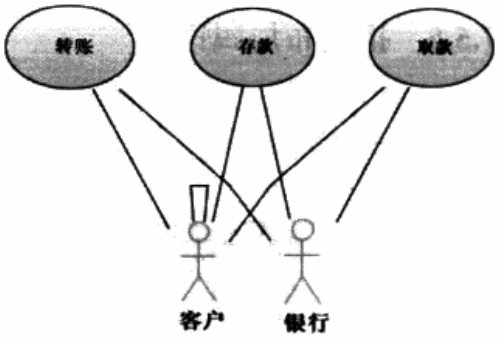


图 17-21 使用分层结构布局

说明：对称结构布局的快捷键为 Ctrl+Alt+Shift+Y。

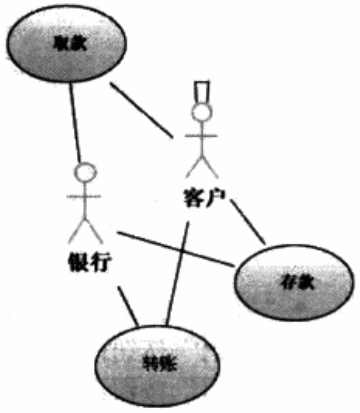


图 17-22 对称结构布局

17.5.3 正交结构布局

正交结构布局将图重新排列为以单个元素为中心的对称结构模式。在 UML 图的空白处单击鼠标右键，选择弹出菜单中的“布局”→“正交结构”命令，或者单击 UML 图编辑器工具栏的“正交结构布局”按钮，都可以指定对称布局。对于如图 17-17 所示的 UML 图，使用对称结构布局后，显示变为如图 17-23 所示。

说明：正交结构布局的快捷键为 Ctrl+Alt+Shift+R。

17.5.4 增量式布局

增量式布局重新排列图，但不改变已经应用于预先存在的图元素的样式。在 UML 图的空白处单击鼠标右键，选择弹出菜单中的“布局”→“增量式”命令，或者单击 UML 图编辑器工具栏的“增量式布局”按钮，都可以指定对称布局。对于如图 17-17 所示的 UML 图，使用对称结构布局后，显示变为如图 17-24 所示。

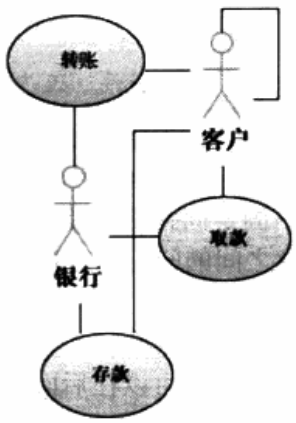


图 17-23 正交结构布局

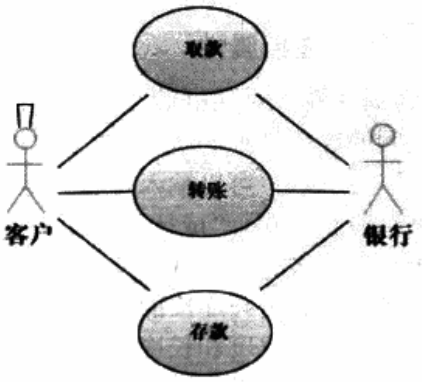


图 17-24 增量式布局

Download at http://www.pin5i.com/

说明：增量式布局的快捷键为 Ctrl+ Shift+I。

17.5.5 指定布局属性

UML 图编辑器中还可以修改布局的属性。在 UML 图的空白处单击鼠标右键，选择弹出菜单中的“布局”→“属性”命令，会打开如图 17-25 所示的“布局属性”对话框。对话框中几个按钮的功能介绍如下。

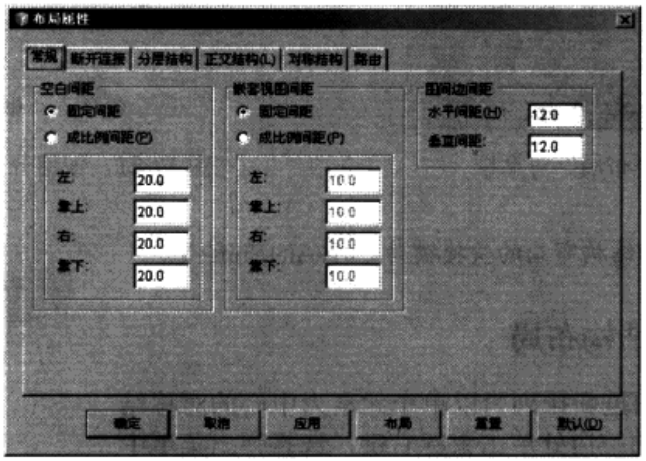


图 17-25 “布局属性”对话框的“常规”标签

- “确定”按钮：为所有适用的图接受指定的内容，然后关闭该对话框。
- “取消”按钮：关闭“布局”属性表单，但不应用任何设置。
- “应用”按钮：应用为所有适用图指定的内容，但不关闭该对话框。
- “布局”按钮：应用为所有适用图指定的内容，但不关闭该对话框，并立即在活动的图编辑器中显示对指定内容所做的任何更改。
- “重置”按钮：将对话框中的所有值重置为打开对话框时显示的值。
- “默认”按钮：将对话框中的所有值重置为安装 NetBeans 集成开发环境时最初指定的值。

1. 常规属性

“布局属性”对话框的“常规”标签如图 17-25 所示，主要用于设置边距。对于空白间距，可以选择“固定间距”单选按钮，然后在下面的四个文本框中指定上下左右的间距大小。“嵌套视图间距”分组框用于设置嵌套视图的间距。在“图间边间距”分组框中，可以设置图之间的水平间距和垂直间距。

2. 断开连接属性

“布局属性”对话框的“断开连接”标签如图 17-26 所示，主要用于指定图元素之间以及节点之间的连接性和高宽比。选中“检测组件”复选框可以检测断开连接的组件（图元素）。“固定间距”文本框中可以输入行和列组件（图元素）之间的间距。固定间距影响使用固定单位表示的所有元素间距。“成比例间距”文本框用于输入行和列元素之间的间距。成比例间距影响与每个元素的大小成比例的所有元素间距。

选中“检测断开连接的节点”复选框将检测断开连接的节点。选中“自动”单选按钮将自动保持图符号中宽度和长度的比例。如果选中“定制”单选按钮，那么可以在后面的文本框中输入定制的宽度和长度比。

Download at http://www.pin5i.com/

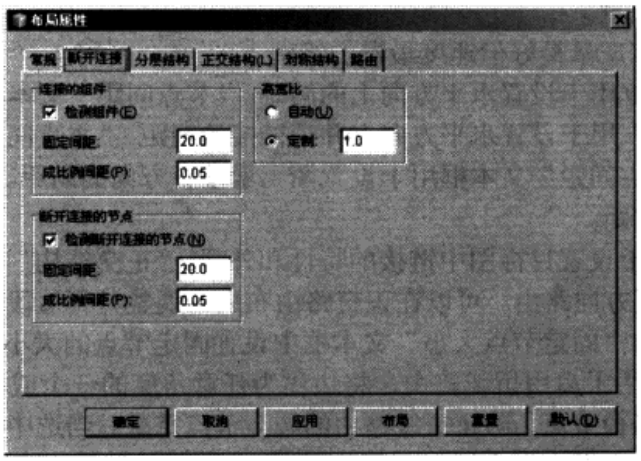


图 17-26 “布局属性”对话框的“断开连接”标签

3. 分层结构属性

“布局属性”对话框的“分层结构”标签如图 17-27 所示，主要为使用分层结构布局的图指定方向、级别对齐、布局质量、间距、路由以及其他参数。

“方向”分组框用于设置图的方向。重新定向图不会自动更改可能依赖于元素特定方面的端口规范、连接器、元素形状或约束。在重新定向复杂图之前，请考虑会产生何种负面影响。可以使用如下的几个方向：

- 从上至下
- 从左至右
- 从下至上
- 从右至左

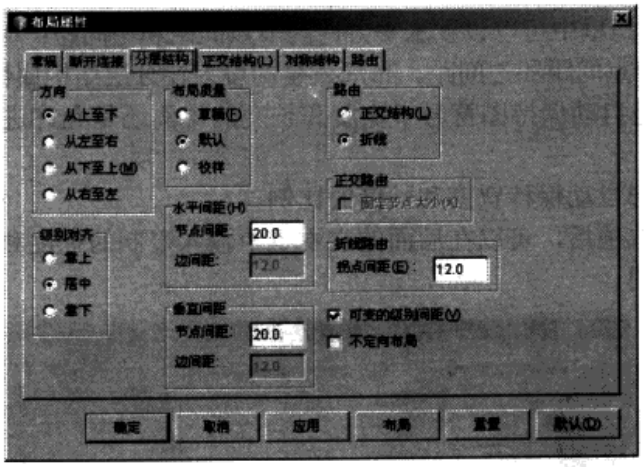


图 17-27 “布局属性”对话框的“分层结构”标签

“级别对齐”分组框用于设置子站点级别在每个树级别上彼此对齐的方式。可以使用如下的几个对齐方式：

- 靠上
- 居中
- 靠下

“布局质量”分组框用于设置生成布局的算法的速度。可以使用如下的几个质量值。

- 草稿：草稿表示质量最差但速度最快的布局级别。

Download at http://www.pin5i.com/

- 默认：默认的质量。
- 校样：校样表示质量最好但速度最慢的级别。

“水平间距”分组框用于设置水平方向上的间距。“节点间距”文本框用于设置图元素的水平方向间距。“边间距”用于设置水平方向上并行链接的间距。“垂直间距”分组框用于设置垂直方向上的间距。“节点间距”文本框用于设置图元素垂直方向的间距。“边间距”用于设置垂直方向上并行链接的间距。

“路由”分组框用于设置目标图中链接的特性和外观。“正交结构”单选按钮，用于启用正交路由，沿水平和垂直方向路由。可以在正交路由布局中调整并行边段的间距。当选中该单选按钮后，可以在下面的“固定节点大小”文本框中设置固定节点的大小。

“折线”单选按钮用于启用折线路由，将边作为任意角度的一个或多个直线段进行路由。该算法根据需要添加路径节点，防止折线路由的边与节点重叠。当选中该单选按钮后，可以在下面的“拐点间距”文本框中指定拐点之间的距离。

“可变的级别间距”复选框用于设置是否要根据级别之间链接的密度，调整一对或相邻级别的间距。

“不定向布局”复选框设置布局是否为无向。这意味着是否将任何链接的源元素和目标元素视为相同。在有向布局图中，链接方向非常重要。网络通常用无向布局表示；而流程通常用有向布局表示。

4. 正交结构属性

“布局属性”对话框的“正交结构”标签如图 17-28 所示，主要为使用正交结构布局的图指定节点间距和边间距、布局质量以及高宽比。

正交结构布局中的间距受以下两种设置的影响：边间距和节点间距。其中，边间距是指水平和垂直方向任意两个相邻平行边之间的最短距离；节点间距是指任意两个相邻元素之间的距离。

在“水平间距”分组框中可以设置水平方向的节点间距和边间距；在“垂直间距”分组框中可以设置垂直方向的节点间距和边间距。布局质量的设置在分层结构属性中已经进行了介绍。

高宽比用于指定是自动保持图符号中宽度和长度的比例还是允许进行定制，可以设置如下的几个值。

- 自动：根据图形自动保持宽度和长度的比例。
- 定制：选中该选项后，可以在后面的文本框中输入定制的宽度和长度的比例。
- 禁用：禁止使用。

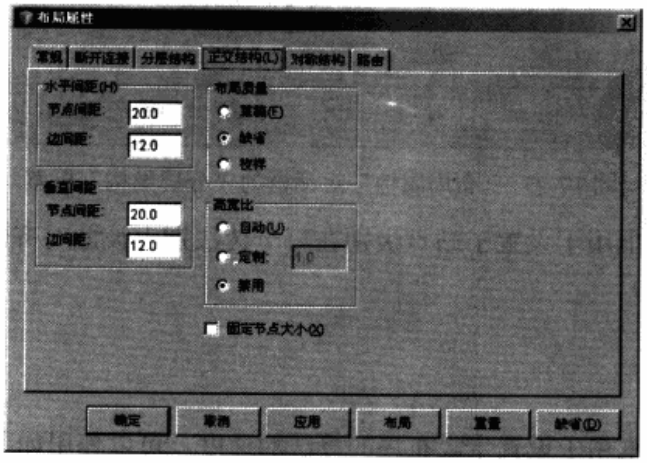


图 17-28 “布局属性”对话框的“正交结构”标签

Download at http://www.pin5i.com/

5. 对称结构属性

“布局属性”对话框的“对称结构”标签如图 17-29 所示，主要为使用对称结构布局的图指定节点间距选项和布局质量。“节点间距”文本框用于设置元素之间的距离。布局质量的设置在分层结构属性中已经介绍过，此处不再赘述。

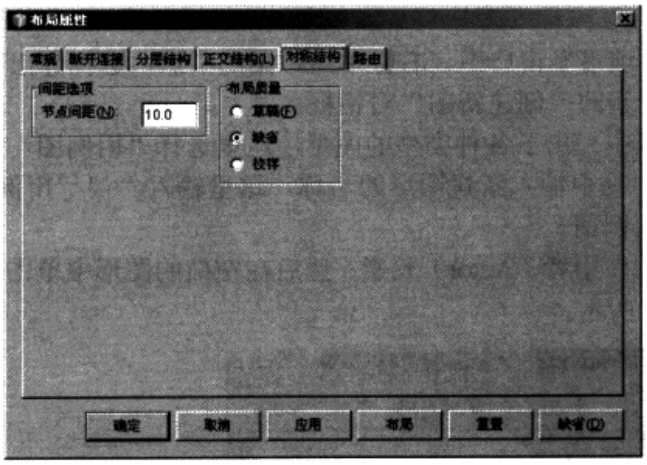


图 17-29 “布局属性”对话框的“对称结构”标签

6. 路由属性

“布局属性”对话框的“路由”标签如图 17-30 所示，主要用于指定水平和垂直间距，以及是否固定节点的大小和位置。

“水平间距”分组框用于设置水平方向的间距；“垂直间距”分组框用于设置垂直方向的间距。其中，“节点间距”文本框用于设置图元素之间的距离；“边间距”文本框用于设置并行链接之间的距离。

“固定节点大小”复选框用于设置链接间距发生的更改时，是否保持元素的大小。“固定节点位置”复选框用于设置当链接发生更改时，是否保持元素的位置。

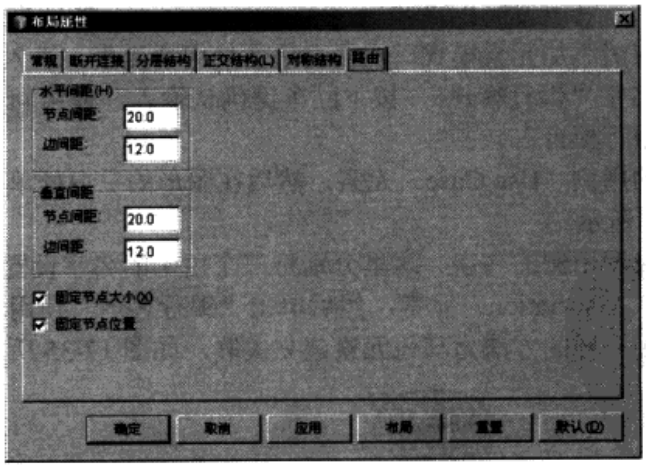


图 17-30 “布局属性”对话框的“路由”标签

17.6 UML 建模实例

在前面几节中介绍了如何 UML 建模的设计环境，本节将通过两个实例演示如何创建用例图和类图。本节的实例都来自一个银行系统模型。

Download at http://www.pin5i.com/

17.6.1 创建用例图实例

用例图用于显示系统行为（从用户角度）、功能描述以及使用系统的用户的图形描述。用例图的主要目的是帮助开发团队以一种可视化的方式理解系统的功能需求分析。用例图是进行系统建模要首先创建的图。

在 NetBeans 的“项目”窗口中，右键单击“图”节点，选择弹出菜单中的“添加图”命令，打开如图 17-31 所示的“创建新图”对话框。

在“图类型”列表中列出了各种类型的图形，这里选择“用例图”节点来创建用例图。在下面的“图名称”文本框中输入新建图形的名称，这里输入“银行用例图”。单击“完成”按钮，创建一个空白的用例图。

在组件面板中单击使用者（Actor）元素，然后在空白的图形中单击鼠标左键两次，添加两个使用者，如图 17-32 所示。

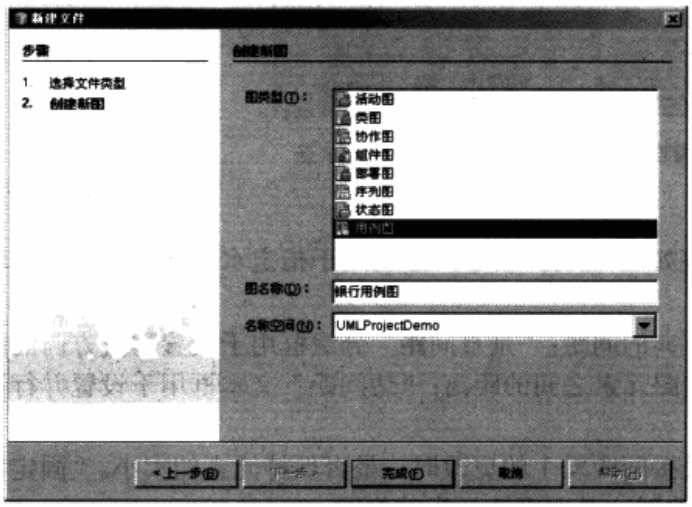


图 17-31 “创建新图”对话框



图 17-32 添加两个使用者

单击鼠标右键，取消添加元素模式。双击左边的使用者元素，进入编辑模式，如图 17-33 所示。输入使用者的名字“银行客户”，按下回车键确认输入。使用同样的方法，把右边的使用者的名字修改为“银行柜台”。

单击组件面板中的用例（Use Case）元素，然后在图形的空白处单击鼠标左键三次，添加三个用例，如图 17-34 所示。

双击用例，可以设置用例的名字。这里分别将三个用例的名字设置为取款、存款和转账。单击组件面板的关联（Association）元素，然后单击“银行客户”使用者，再单击“取款”用例，建立一个关联。用同样的方法为其他元素建立关联，如图 17-35 所示。



图 17-33 修改使用者的名字

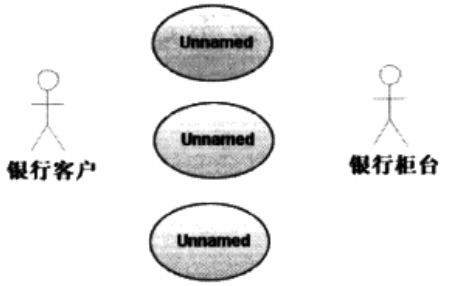


图 17-34 添加三个用例

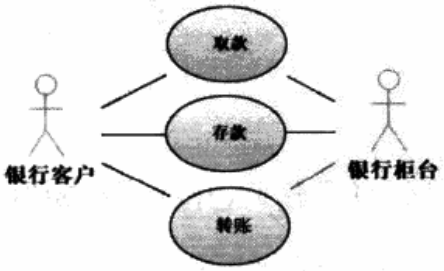


图 17-35 为元素建立关联

Download at http://www.pin5i.com/

至此，创建了一个简单的用例，表示了银行客户要在银行中进行的三个基本用例，即取款、存款和转账。

17.6.2 创建类图实例

类图是系统中使用到的类的描述和设计，这也是 UML 设计的核心模型。设计人员必须在类图中定义系统中类的名字、类中的属性和操作，以及类和类之间的关系。

在 NetBeans 的“项目”窗口中，右键单击“图”节点，选择弹出菜单中的“添加图”命令，打开如图 17-36 所示的“创建新图”对话框。

在“图类型”列表中列出了各种类型的图形，这里选择“类图”节点来创建类图。在下面的“图名称”文本框中输入新建图形的名称，这里输入“银行类图”。单击“完成”按钮，创建一个空白的类图。

单击组件面板中的类（Class）元素，然后在图形的空白处单击鼠标左键，添加一个类元素，如图 17-37 所示。单击鼠标右键，取消添加模式。双击该元素，输入该类的名字，这里输入“Custom”，如图 17-38 所示。

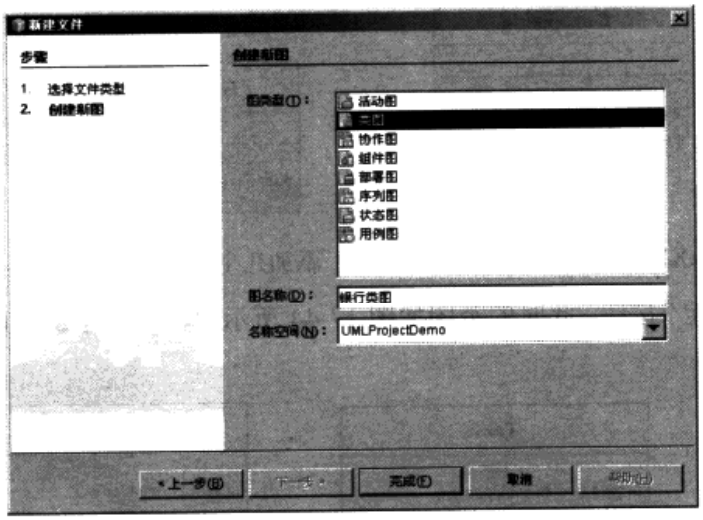


图 17-36 “创建新图”对话框

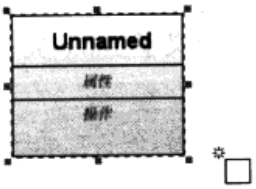


图 17-37 添加一个类元素

单击元素的“属性”区域，选择弹出菜单中的“插入属性”命令，插入一个属性，如图 17-39 所示。输入属性的名字后，可以把光标向左移动，修改属性的类型和可视性。当光标停留在这两个类型上时，同时按下 Ctrl 和向下键，会自动打开上下文菜单供用户选择。图 17-40 是属性类型的提示菜单。

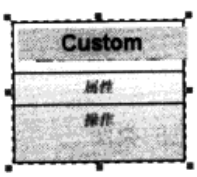


图 17-38 修改类名字

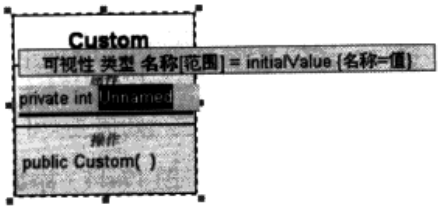


图 17-39 插入属性

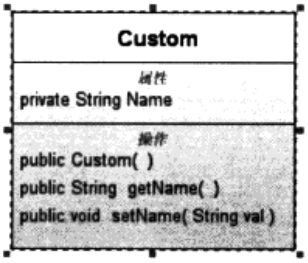


图 17-40 属性类型的上下文菜单

Download at http://www.pin5i.com/

输入完属性后，按下回车键结束输入。此时，会自动添加两个 getXXX 和 setXXX 操作，如图 17-41 所示。

用同样的方法可以添加如图 17-42 所示的几个类。然后为这几个类添加关系。现金柜台是继承了银行柜台，为这两个元素添加继承关系。在组件面板中选中泛化（Generalization）元素，单击现金柜台元素，再单击银行柜台元素，这样就为两个元素建立了关系。此时，会打开如图 17-43 所示的“选择要重定义的方法”对话框。



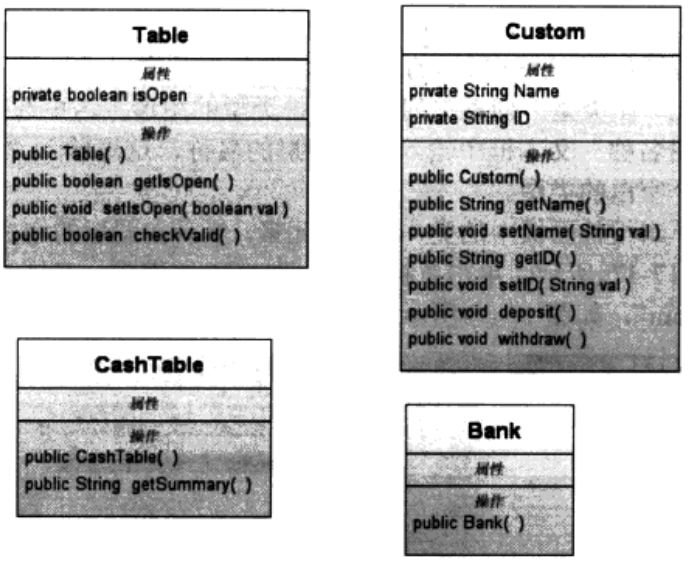


图 17-41　自动为属性添加 getXXX 和 setXXX 操作　　图 17-42　添加几个类

这里选择所有的方法，单击“确定”按钮，此时的类图如图 17-44 所示。可以看到 CashTable 类中重定义了 Table 类中的三个方法。



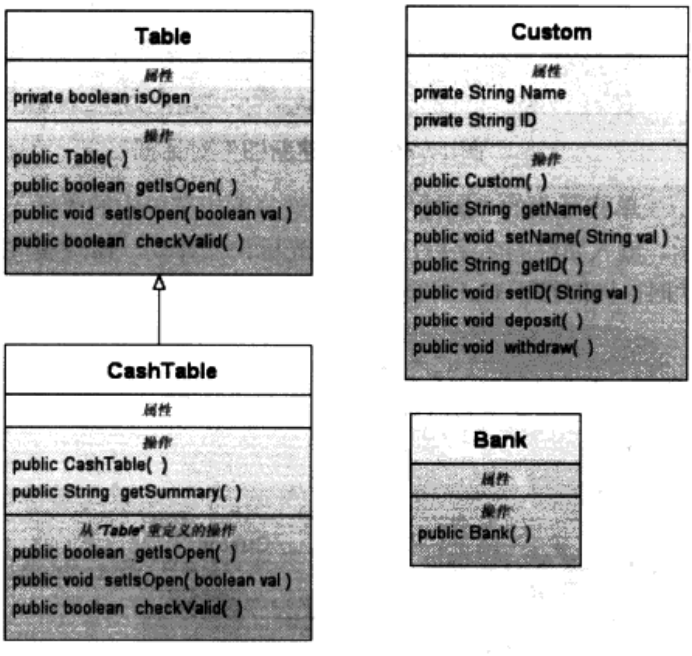


图 17-43　“选择要重定义的方法”对话框　　图 17-44　添加了继承关系的类图

在一个银行中，可能会包含一个或者多个银行柜台，因此银行类和现金柜台类之间可以用

Download at http://www.pin5i.com/

关联关系来表示。单击组件面板中的关联（Association）元素，再分别单击银行类和现金柜台类。右键单击靠近银行类一侧的关联关系，选择弹出菜单中的“设置多重性”→“1”命令，添加一个多重性关系。右键单击靠近银行柜台类一侧的关联关系，选择弹出菜单中的“设置多重性”→“1..*”命令，添加一到多的关系。

添加了继承和关联关系的类图如图 17-45 所示。

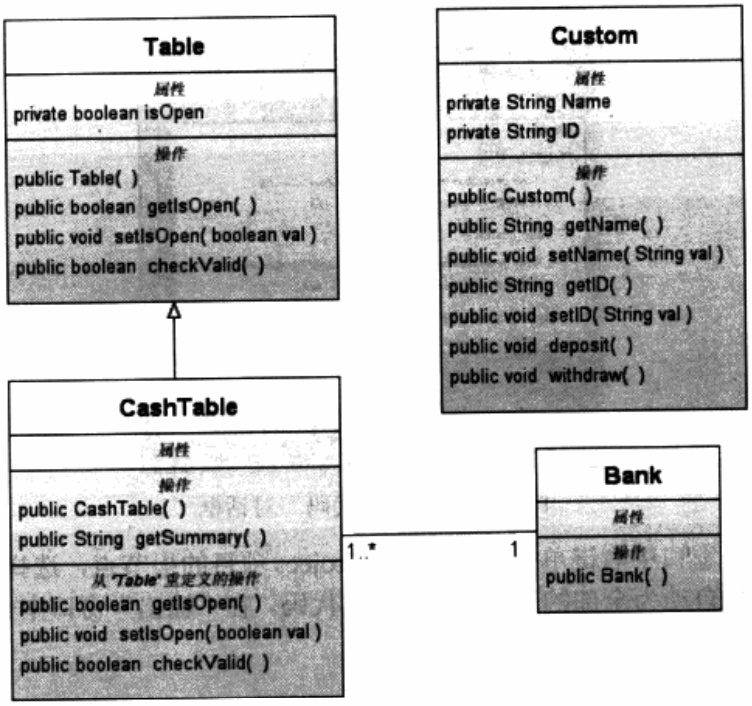


图 17-45　添加继承和关联关系的类图

17.7　自动生成代码和反向工程

NetBeans 中设计的 UML 类图可以自动生成代码，对于已经存在的源代码还可以进行逆向工程反向创建 UML 模型。

17.7.1　自动生成代码

在自动生成代码前，首先要创建一个项目，例如这里创建一个标准的 Java 应用程序，程序的名字为 UMLDemoProjectSrc。

在“项目”窗口中右键单击 UML 项目节点，选择弹出菜单中的“生成代码”命令，打开“生成代码”对话框，如图 17-46 所示。

在“目标项目”中选择刚刚创建的 UMLDemoProjectSrc 项目。如果要生成代码是备份现有的文件，则要选中“备份现有的源文件”复选框。在“模板”列表中选择一个使用的模板。单击“确定”按钮，在“输出”窗口显示如下的信息，表示自动创建成功。

```
开始处理 生成代码：1 项

代码生成选定的选项：
  源文件夹位置 C:\Documents and Settings\Robin\桌面\UMLProjectDemoSrc\src

处理第 1 个元素（共 1 个）：Class Table
```

Download at http://www.pin5i.com/


```
==================================
任务成功 （总时间：0 秒）
```

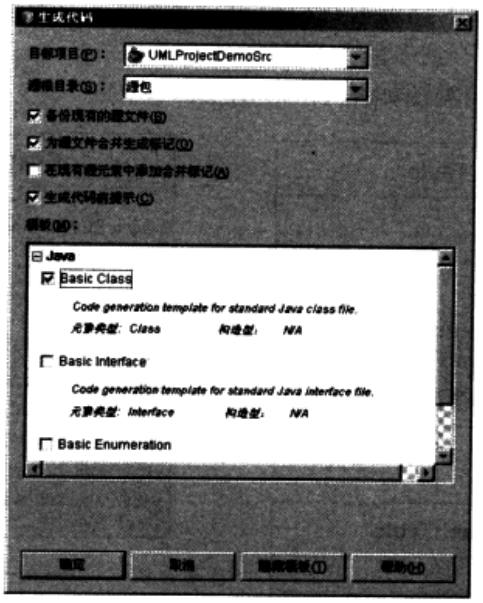


图 17-46 “生成代码”对话框

如果在“项目”窗口中右键单击 UMLDemoProject 项目的根节点，选择弹出菜单中的“生成代码”命令，则会自动为全部的 UML 类图生成代码。在“输出”窗口中会显示如下的信息。

```
开始处理 生成代码：4 项

代码生成选定的选项：
  源文件夹位置 C:\Documents and Settings\Robin\桌面\UMLProjectDemoSrc\src

处理第 1 个元素（共 4 个）：Class Table
处理第 2 个元素（共 4 个）：Class Custom
处理第 3 个元素（共 4 个）：Class CashTable
处理第 4 个元素（共 4 个）：Class Bank

==================================
任务成功 （总时间：1 秒）
```

此时，选择 UMLProjectDemoSrc 项目，打开 CashTable 类文件。可以看到，自动创建了如下的代码。

```
public class CashTable extends Table {
    public CashTable() {
    }

    public String getSummary() {
        return null;
    }

    public boolean getIsOpen() {
        return true;
    }

    public void setIsOpen(boolean val) {
    }

    public boolean checkValid() {
```

Download at http://www.pin5i.com/


```
            return true;
        }
    }
```

17.7.2 对单个源文件进行逆向工程

NetBeans 可以将 Java 源代码逆向工程为 UML 建模项目，该项目包含了源代码的全部结构信息（不含注释和空白）。使用 UML 建模对 Java 代码进行逆向工程时，NetBeans 将执行以下操作：

- 在 UML 模型中生成类元素以反映 Java 源代码的类和接口；
- 将 Java 类成员和方法作为其各自类组件内的属性和操作添加到 UML 模型中；
- 通过 Java import 语句派生出文件的依赖关系；
- 生成 UML 建模项目并在“项目”窗口中放置新的 UML 项目节点。

现在对第 17.7.1 节中的 CashTable 类进行修改，添加如下的 getAmount 方法。

```
public class CashTable extends Table {
    public CashTable() {
    }

    public String getSummary() {
        return null;
    }

    public boolean getIsOpen() {
        return true;
    }

    public void setIsOpen(boolean val) {
    }

    public boolean checkValid() {
        return true;
    }

    public double getAmount() {
        return 0;
    }
}
```

在“项目”窗口中，右键单击 UMLProjectDemoSrc 项目下的“源包”→“<默认包>”→“CashTable.java”节点，选择弹出菜单中的“逆向工程”命令，打开“逆向工程”对话框，如图 17-47 所示。

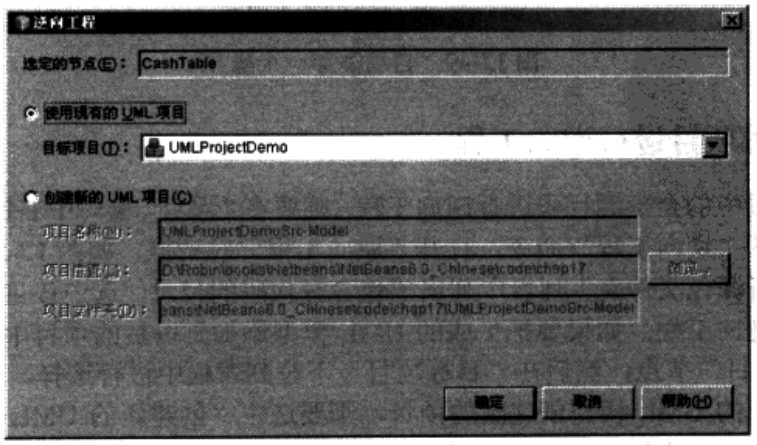


图 17-47 “逆向工程”对话框

Download at http://www.pin5i.com/

这里选中“使用现有的 UML 项目”单选按钮，然后在“目标项目”下拉列表框中选择项目 UMLProjectDemo。单击“确定”按钮，由于该元素已经存在，所以会显示如图 17-48 所示的“模型元素覆盖授权”对话框。

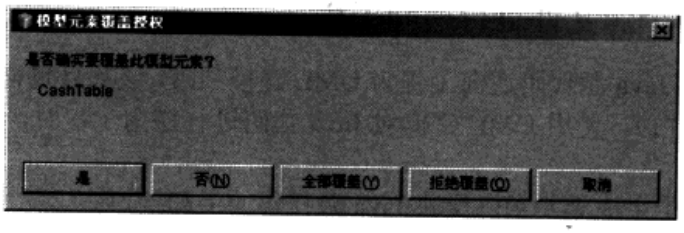


图 17-48 “模型元素覆盖授权”对话框

单击“是”按钮，确认覆盖原来的元素。在“输出”窗口中会显示如下的信息，表示逆向工程成功。

```
开始处理 逆向工程

解析 1 元素

分析 1 符号的属性与操作

解析 5 属性类型

集成 1 元素

生成查询缓存。

================================
任务成功 （总时间：2 分钟 17 秒）
```

此时，打开 CashTable 模型，会看到自动添加了一个操作，如图 17-49 所示。

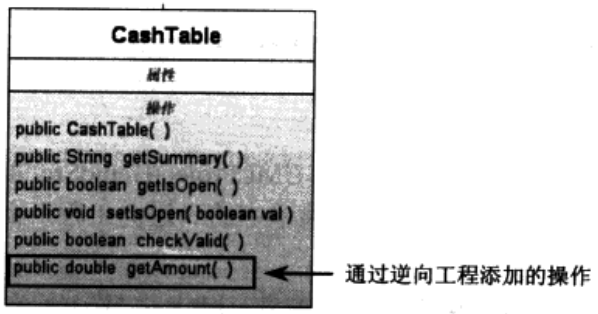


图 17-49 自动添加一个操作

17.7.3 对项目进行逆向工程

如果要把项目中的全部源代码进行逆向工程，则要在“项目”窗口中右键单击项目根节点，选择弹出菜单中的“逆向工程”命令，打开“逆向工程”对话框，如图 17-50 所示。

在“Java 项目源包文件夹”列表中列出了包含源代码的目录，选中该目录后将把目录中的全部源代码进行逆向工程。如果要把生成的 UML 模型添加到以后的项目中，则需要选中“使用现有的 UML 项目”节点，然后在“目标项目”下拉列表框中进行选择。

如果要为生成的 UML 模型创建新的项目，则要选中“创建新的 UML 项目”单选按钮，然后在“项目名称”文本框中指定新项目的名称；在“项目位置”文本框中输入项目的位置。

Download at http://www.pin5i.com/

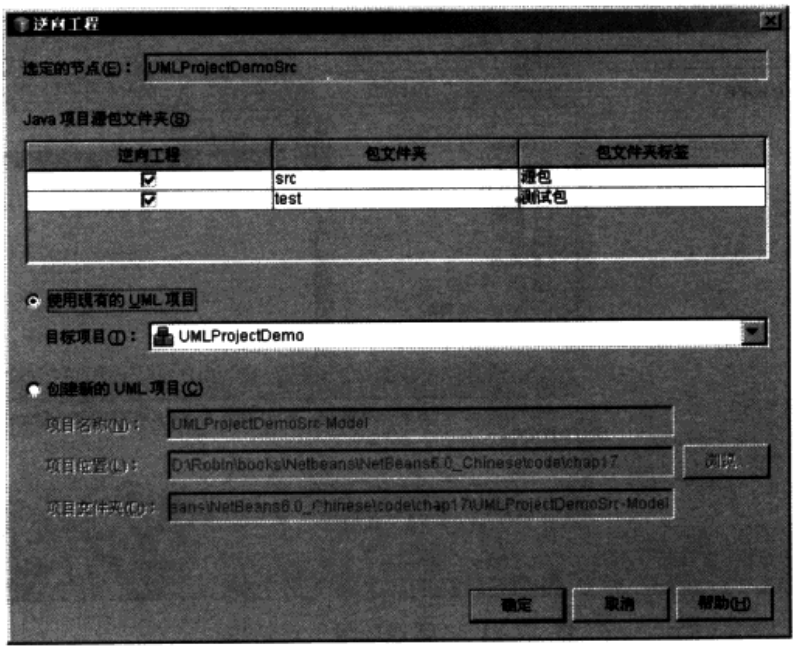


图 17-50 “逆向工程”对话框

单击“确定”按钮。如果模型已经存在，会显示如图 17-48 所示“模型元素覆盖授权”对话框。单击“全部覆盖”按钮，会在“输出”窗口中显示如下的信息，表示逆向工程成功了。

```
开始处理 逆向工程

解析 4 元素

分析 4 符号的属性与操作

解析 5 属性类型

集成 4 元素

生成查询缓存。

================================
任务成功 （总时间：54 秒）
```



说明： UML 建模功能在逆向工程或代码生成期间不使用代码标记，而其他逆向工程工具则包含标记和不必要的注释，它们尽管不参与编译，但可能会造成源代码混乱无序。UML 建模功能可以使得执行逆向工程前后的代码保持一致。



17.7.4 使用向导进行逆向工程

在 NetBeans 集成开发环境中，选择“文件”→“新建项目”命令，打开如图 17-51 所示的“新建项目”对话框。

在“类别”列表中选择“UML”节点，然后在右边的“项目”列表中选择“逆向工程后的 Java 平台模型”节点。单击“下一步”按钮，打开“新建逆向工程后的 Java 平台模型”对话框，如图 17-52 所示。

Download at http://www.pin5i.com/

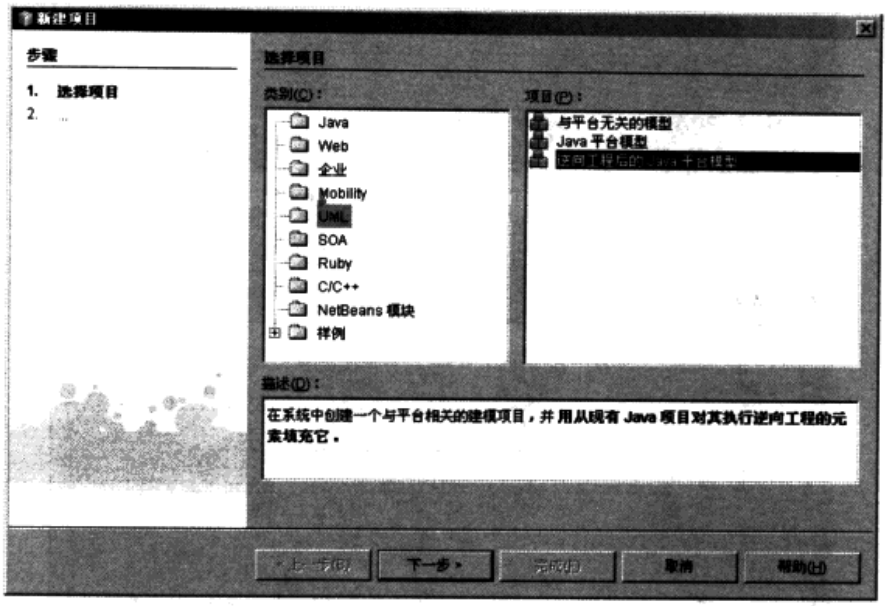


图 17-51 “新建项目”对话框

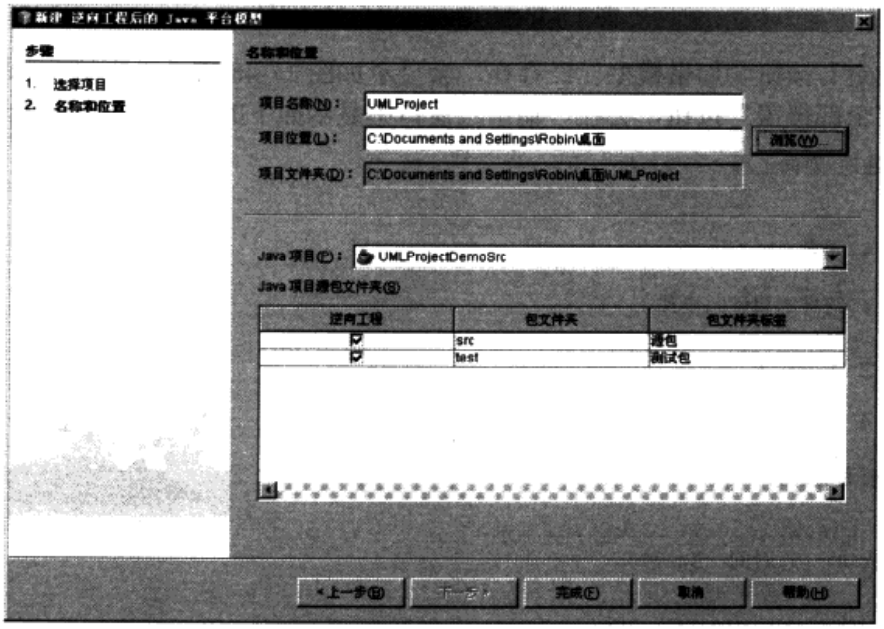


图 17-52 “新建逆向工程后的 Java 平台模型”对话框

在“项目名称”文本框中输入新的项目的名字，在“项目位置”文中框中输入项目的位置，或者单击“浏览”按钮进行选择。

在“Java 项目”下拉列表框中选择要进行逆向工程的项目，在“Java 项目源包文件夹”列表中选择要进行逆向工程的目录。单击“完成”按钮，开始进行逆向工程。此时，会在新的项目中创建所有的模型。

17.8 UML 建模的其他高级操作

前面几节已经介绍了 UML 建模的设计环境和基本的建模方法，本节将主要讨论一些 UML 建模中的高级操作，例如查找/替换、关联元素、过滤元素等。

Download at http://www.pin5i.com/

17.8.1 查找和替换

在 UML 项目中支持各种方式的查找和替换。在“项目”窗口中右键单击 UML 项目根节点，选择弹出菜单中的“查找”命令，可以打开“查找”对话框，如图 17-53 所示。

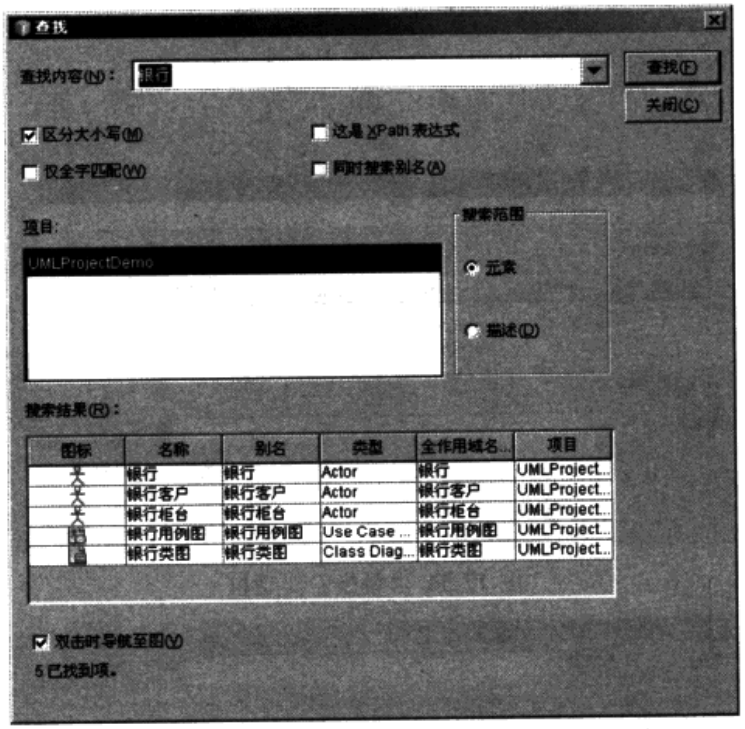


图 17-53 “查找”对话框

在“查找内容”文本框中输入要查找的内容，单击“查找”按钮即可进行查找。查找到的结果将显示在“搜索结果”列表中。在“项目”窗口中列出了可以进行查找的项目。选中“元素”单选按钮，则只在元素中进行查询；如果选中“描述”单选按钮，则在描述中进行查找。

下面几个复选框可以指定查找的参数。

- “区分大小写”复选框：查找到时区分大小写。
- “仅全字匹配”复选框：只有被查找内容（元素或者描述）和查找内容一致时才认为查找到目标。选中该模式后，将进行精确查询。
- “这是 XPath 表达式”复选框：表示查找的内容是 XPath 表达式。
- “同时搜索别名”复选框：查找时也搜索别名。

在“项目”窗口中右键单击 UML 项目根节点，选择弹出菜单中的“替换”命令，可以打开“替换”对话框，如图 17-54 所示。

“替换”对话框和“查找”对话框类似，只是在下面多了一个“替换为”下拉列表框和两个按钮。在查找到结果后，可以在“替换为”下拉列表框输入或者选择要替换的内容，单击“替换”按钮可以替换单个选中的查找结果，单击“全部替换”则替换全部的内容。

17.8.2 关联元素

NetBeans 的 UML 建模功能包括关联功能，通过“关联”对话框可以将任意建模元素与其他任何元素相关联。

右键单击要关联的元素，选择弹出菜单中的“关联”命令，会打开“关联”对话框，如图

Download at http://www.pin5i.com/

17-55 所示。“关联”对话框类似于“查找”对话框。首先要查找被关联的内容，查找到结果后可以选择某一个结果，然后单击“关联”按钮进行关联。如果单击“全部关联”按钮，则会关联全部的内容。

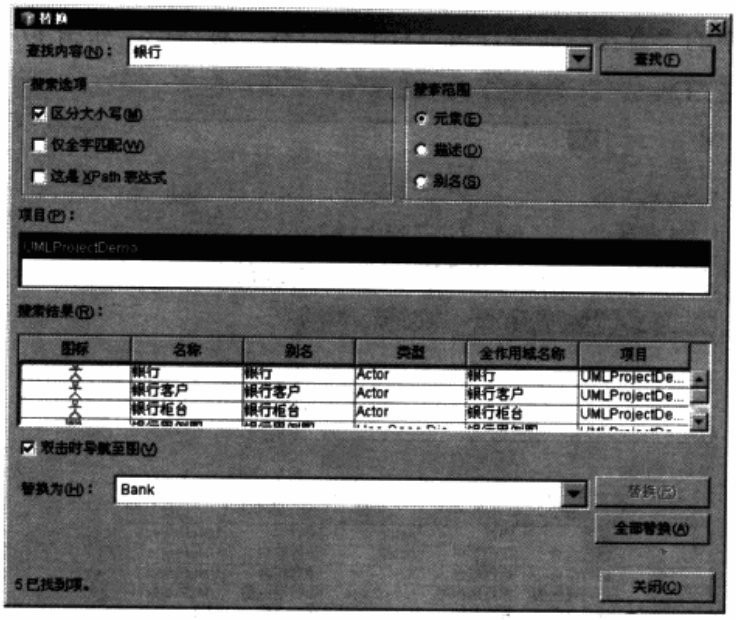


图 17-54 “替换”对话框

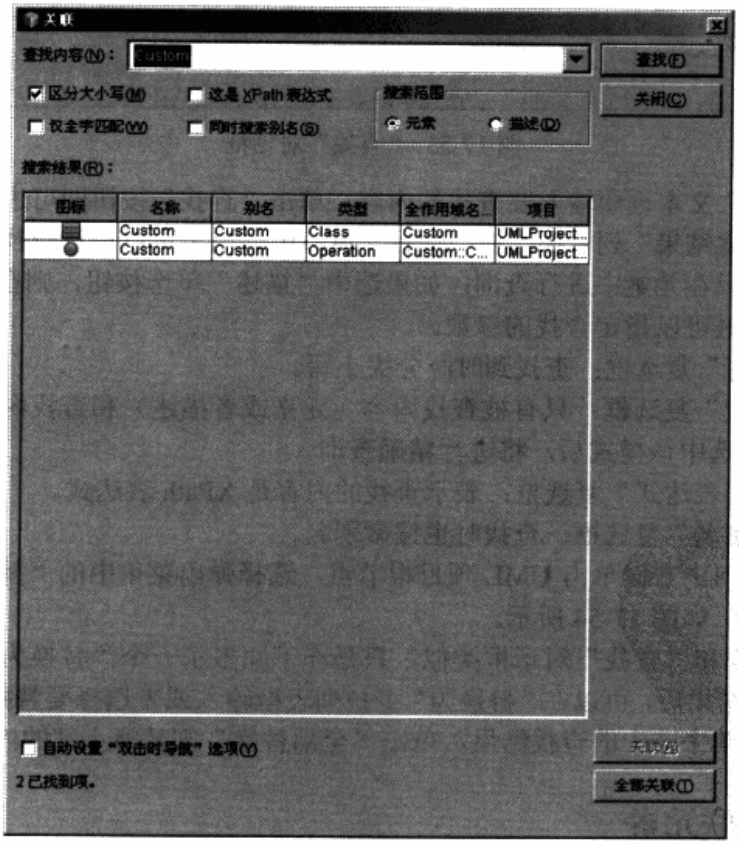


图 17-55 “关联”对话框

NetBeans 将关联元素，并在“项目”窗口和“属性”窗口中创建一个引用，分别如图 17-56 和图 17-57 所示。

Download at http://www.pin5i.com/

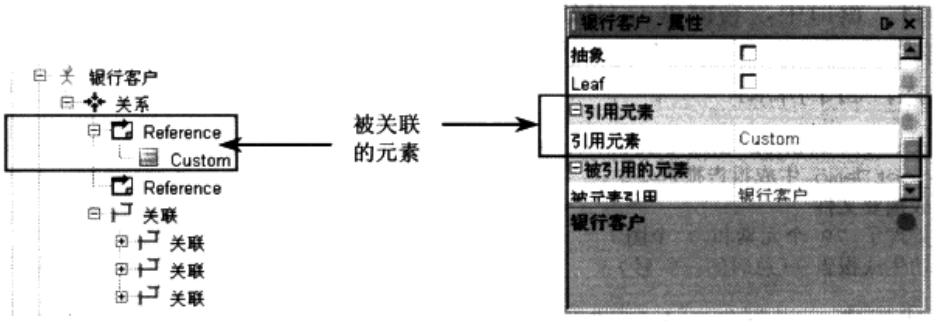


图 17-56 “项目”窗口中被关联的元素　　图 17-57 “属性”窗口中被关联的元素

17.8.3 过滤元素类型

通过“项目”窗口过滤元素类型，可以暂时隐藏“项目”窗口中的元素节点。例如，如果展开的项目树包含此时不需要看到的许多使用者，则可以通过过滤掉“使用者”节点来节省空间。如果要使项目树的外观平整，则可以过滤掉“包”节点。将过滤器应用于“项目”窗口后，所有元素类型仍显示在图编辑器和其他窗口中。

在“项目”窗口中打开项目，右键单击“模型”节点，然后选择“过滤器”命令，将打开“过滤器折叠节点警告”对话框，如图 17-58 所示。

选中“不再询问”复选框，则下次将不再出现该对话框。单击“是”按钮，打开“过滤器”对话框，如图 17-59 所示。

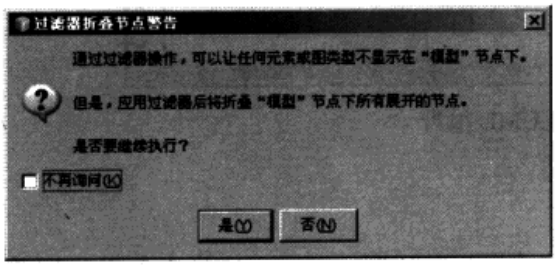


图 17-58 “过滤器折叠节点警告”对话框

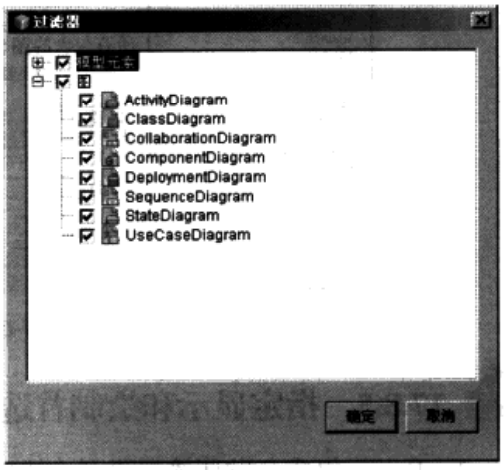


图 17-59 “过滤器”对话框

展开“模型元素”节点，然后选择要过滤掉的元素类型。复选标记指示不被过滤的元素类型。单击某个元素类型时，复选标记将消失。再次单击该元素类型时，复选标记又会出现。同样，展开“图”节点可以选择要过滤的图。

单击“确定”按钮，NetBeans 会将过滤器应用于“项目”窗口。现在，“项目”窗口将显示除选择要过滤掉的元素类型之外的所有元素类型。

17.8.4 UML 模型报告

UML 建模提供了一个报告功能，利用该功能可生成一个详细介绍有关 UML 项目重要信息的报告。该报告是只读的，并且以 Javadoc 格式生成。它包含了有关 UML 项目的包、类、接口以及其他元素的信息。

Download at http://www.pin5i.com/

在“项目”窗口中，右键单击 UML 项目，从弹出式菜单中选择“生成模型报告”命令。NetBeans 将在后台生成。“输出”窗口将打开一个名为“UML 报告日志”的标签，它会显示报告生成的状态，如下所示：

```
    在 C:\Documents and Settings\Robin\桌面\UMLProjectDemo\report 处为
UMLProjectDemo 生成报告报告
    生成摘要文件
    已处理了 79 个元素和 3 个图
    成功生成报告 （总时间：9 秒）
```

一旦生成报告，则会在 Web 浏览器中显示该报告，如图 17-60 所示。



图 17-60 生成的 UML 报告

17.8.5 指定显示和绘制首选项

NetBeans 集成开发环境提供了“显示和绘制首选项”对话框，可以用来指定如何处理单个“UML 图”窗口中的缩放、透视、分辨率和类似的显示问题。在 UML 图空白处单击鼠标右键，选择弹出菜单中的“属性”命令，可以打开“显示和绘制首选项”对话框，如图 17-61 所示。图中各个参数的说明如表 17-1 所示。

表 17-1 “显示和绘制首选项”对话框中各个参数说明

参 数	说 明
自动隐藏滚动条	可以指定显示图全貌时是否要隐藏或显示“UML 图”窗口滚动条。选中该复选框会自动隐藏不必要的滚动条，这样可以在“UML 图”窗口中获得更大的图像空间
显示工具提示	控制显示或隐藏工具提示
最大缩放百分比	用于指定最大比例，若超出该比例，NetBeans 集成开发环境将无法再进行放大
最小缩放百分比	用于指定最小比例，若低于该比例，NetBeans 集成开发环境将无法再进行缩小
撤销堆栈大小	输入要撤销的级别数

Download at http://www.pin5i.com/

续表

参　数	说　明
压缩元边	用于指定在折叠操作中是否将多个边合并为一个元边
几何调整	用于指定在展开、折叠、取消折叠、隐藏和取消隐藏命令完成其操作后是否调整绘图内对象的间距
部分选择	用于指定是否选中部分存在于选择区域选择矩形内的对象
不透明移动	用于指定拖动时是绘制整个对象还是只绘制轮廓
边命中容错	用于指定命中测试的假定的边宽
重新连接边灵敏度	在确定何时激活重新连接边工具时，该选项用于指定阈值
交互缩放灵敏度	用于指定交互缩放时的鼠标灵敏度
平移灵敏度	用于指定平移时的鼠标灵敏度
行程时间	用于指定在链接导航中从某个端点节点到下一个端点节点所需的时间
闪烁时间	用于指定在处理边时边闪烁的时间
帧数	用于指定从边的一端滚动到另一端所使用的帧数（或步骤数）

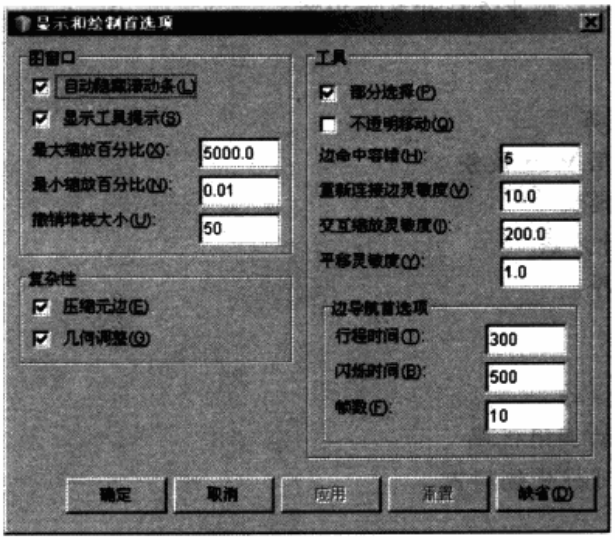


图 17-61 “显示和绘制首选项”对话框

17.9 小结

本章主要介绍如何在 NetBeans 集成开发环境中进行 UML 建模。为了能够使用建模功能，需要首先安装 UML 建模模块。本章的介绍是基于读者有一定的 UML 知识的，因此在学习本章前读者需要首先具备一定的基础知识。由于本章的内容相对独立，如果不需要使用 UML 建模功能，读者可以跳过本章，接续下面章节的学习。通过本章的学习，读者应该掌握以下基本内容：

- 如何安装 UML 建模模块
- UML 建模设计环境
- 如何创建简单的用例图和类图
- 如何自动生成代码和进行逆向工程
- 如何生成 UML 模型报告

Download at http://www.pin5i.com/

第 18 章

Web 服务以及 SOA 程序开发

Web 服务是分布式应用程序组件，它们符合一定的标准，因而可供外部使用。Web 服务可以独立开发，并在不同的系统中运行，因此可以解决各种计算机应用程序的集成问题。面向服务的体系结构 SOA（Service-Oriented Architecture）主要用于组装 Java 业务集成 JBI（Java Business Integration）服务器的部署单元。在复合应用程序项目中，可以使用多种项目类型（例如 BPEL、XSLT 等）组装应用程序。

本章首先介绍 Web 服务的开发方法。然后通过实例介绍了 SOA 程序的简单方法，其中包括如何进行可视化的 BPEL 商业流程开发等内容。

本章主要内容

- 如何创建 Web 服务
- 如何在客户端使用 Web 服务
- 如何创建 BPEL 模块
- BPEL 流程的设计方法
- 如何集成复合程序

本章亮点

- 如何为服务添加操作
- 自动添加对服务的调用代码
- 如何使用 BPEL 可视化编辑器

18.1 Web 服务开发

Web 服务是除了 Java 桌面程序、Web 应用程序、企业应用程序以外的另外一种程序类型。它可以通过开放式网络将某个应用程序的全部或部分功能向其他应用程序公开，以提供必要的服务。例如某航空公司，可以提供查询机票的预定情况等。

18.1.1 Web 服务的类型以及规范

NetBeans 集成开发环境支持两种类型的编程模型：JAX-RPC 和 JAX-WS。其中，基于 XML 的远程过程调用 JAX-RPC（Java API for XML-based Remote Procedure Call）在 JSR 101 规范中定义。JAX-RPC 能够使服务器使用标准的 API 定义其服务并且能够用 WSDL 描述其服务。在客户端，用户可以使用这个标准的 API 与服务器进行通信。

JAX-WS（Java API for XML Web Services）作为 JAX-RPC 的后续发行版本，简化了使用 Java 技术开发 Web 服务的工作。例如，它通过提供标注来简化 Web 服务开发并减小运行时 JAR 文件的大小，以解决 JAX-RPC 1.1 中的某些问题。

无论是 JAX-RPC，还是 JAX-WS 编程模型，都会使用到 SOAP、WSDL 和 UDDI 这三个

Download at http://www.pin5i.com/

规范。简单对象访问协议 SOAP（Simple Object Access Protocol）定义了调用 Web 服务所使用的机制以及数据的返回方式。

Web 服务定义语言 WSDL（Web Service Definition Language）描述了 Web 服务的外部接口。WSDL 文件是有关如何交换数据和消息以及如何发送和接收这些消息的一种协议。

通用发现、描述及集成 UDDI（Universal Discovery, Description, and Integration）中包含 Web 服务的相关信息，例如包括 WSDL 文件及运行服务的位置。



18.1.2 创建一个简单的 Web 服务

本节将创建一个简单的 Web 服务，用于完成特定的数学计算。然后再创建一个客户端使用这个服务。

新建一个名为 WebServiceDemo 的 Web 应用程序，并设置为主项目。选择“文件”→“新建文件”命令，打开“新建文件”对话框。在“类别”列表中选择“Web 服务”节点，在“文件类型”列表中选择“Web 服务”节点。

单击“下一步”按钮，打开“新建 Web 服务”对话框，如图 18-1 所示。在“Web 服务名字”中输入服务的名字，在客户端访问服务时要使用到该名字。这里输入 MyWebService。在“包”下拉列表框中选择已经有的包，或者输入新包的名字。这里输入“myservice”。

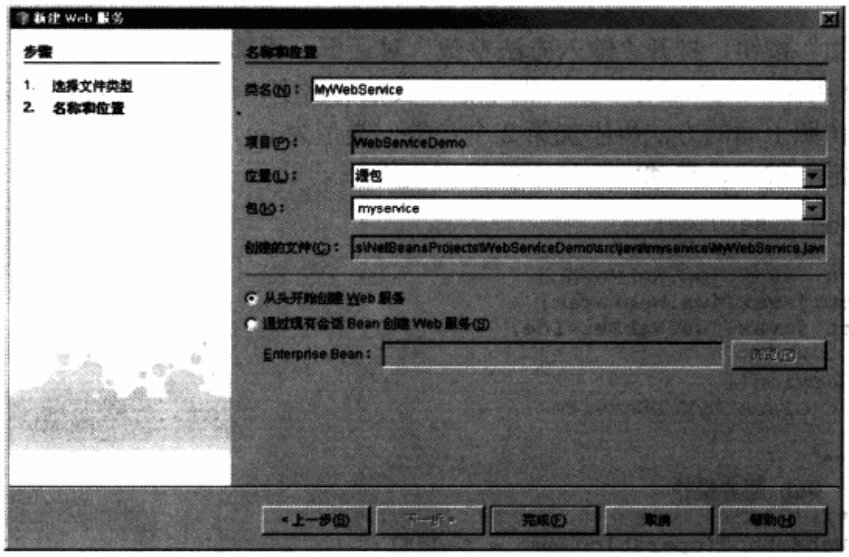


图 18-1 “新建 Web 服务”对话框

选中“从头开始创建 Web 服务”单选按钮，将创建一个空白的服务。在没有要公开的代码时，可以使用这种类型。如果 EJB 模块中有要公开的会话 Bean，或者 Web 应用程序中有要公开的类，需要选中“通过现有会话 Bean 创建 Web 服务”单选按钮。单击“浏览”按钮，可以选择要作为 Web 服务公开的会话 Bean。

单击“完成”按钮，将创建一个 MyWebService 类，代码如下所示：

```
package myservice;

import javax.jws.WebService;

@WebService()
public class MyWebService {

}
```

Download at http://www.pin5i.com/

这里使用了@WebService 注入。在源代码编辑器中 MyWebService 类上单击鼠标右键，选择弹出菜单中的“Web 服务”→“添加操作”命令，打开“添加操作”对话框，如图 18-2 所示。

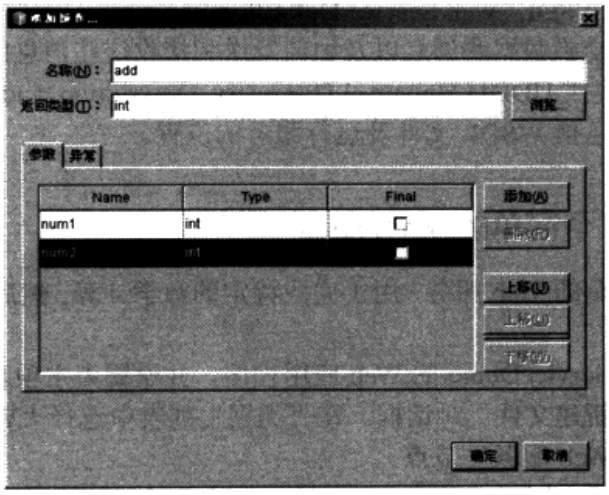


图 18-2 “添加操作”对话框

在“名称”文本框中输入操作的名字，这里输入“add”。在“返回类型”文本框中输入 int 类型。

单击“添加”按钮，打开“输入方法参数”对话框，添加第 1 个参数。在“类型”下拉列表框中选择 int 类型，在“名称”文本框中输入参数的名字，这里输入 num1。单击“确定”按钮添加参数。仿照上面的方法再添加第 2 个参数，类型为 int，参数名为 num2。

单击“确定”按钮，自动添加如下的操作代码：

```
package myservice;

import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;

@WebService()
public class MyWebService {

    /**
     * Web 服务操作
     */
    @WebMethod(operationName = "add")
    public int add(
            @WebParam(name = "num1") int num1,
            @WebParam(name = "num2") int num2) {
        //TODO 在此处编写实现代码:
        return 0;
    }
}
```

把代码修改为返回两个参数 num1 和 num2 的和，代码如下所示：

```
    public int add(
            @WebParam(name = "num1") int num1,
            @WebParam(name = "num2") int num2) {
        //TODO 在此处编写实现代码:
        return num1 + num2;
    }
```

单击源代码编辑器工具栏上的“设计”按钮，打开设计视图，在视图中可以看到刚刚添加的 add 操作，如图 18-3 所示。

Download at http://www.pin5i.com/

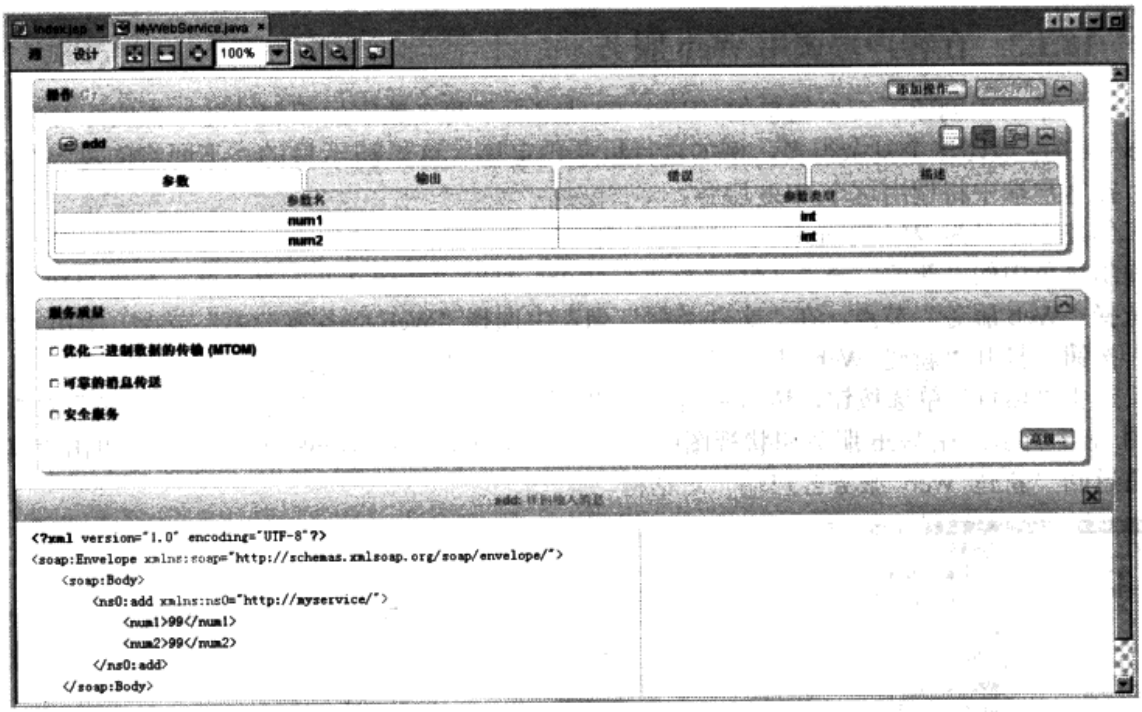


图 18-3 服务的“设计”视图

至此，添加了一个 Web 服务的操作。在客户端就可以访问这个操作，以实现某种功能。在“项目”窗口中右键单击项目节点，选择弹出菜单中的“取消部署后再次部署”命令，把 Web 服务部署到服务器。在“项目”窗口中右键单击项目节点下的“Web 服务”→“MyWebService”节点，选择弹出菜单中的“测试 Web 服务”命令，打开如图 18-4 所示的测试界面。

在文本框中输入任意两个数，单击“add”按钮。该页面将调用刚刚创建的 Web 服务进行计算。测试结果如图 18-5 所示。

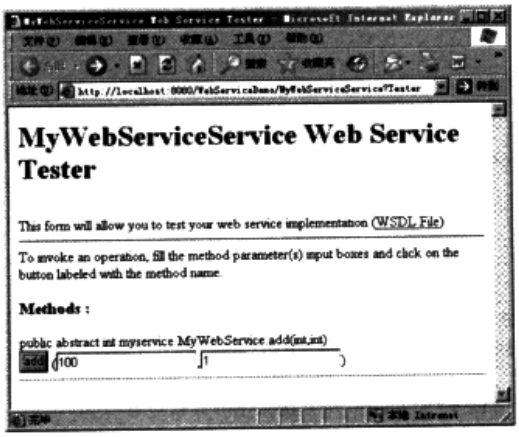


图 18-4 Web 服务的测试界面

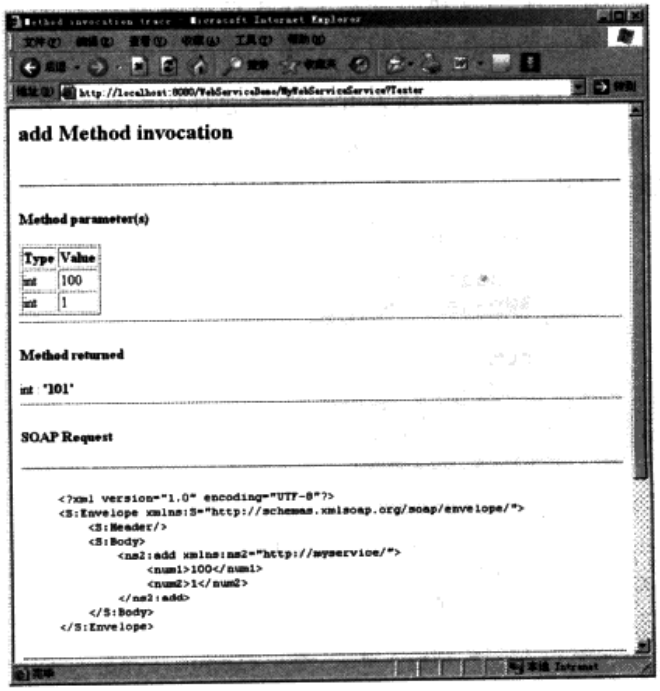


图 18-5 Web 服务的测试结果

Download at http://www.pin5i.com/

18.1.3 在客户端调用 Web 服务

上一小节中创建了一个简单的 Web，即只完成了两个数字的相加功能。在实际开发中，Web 服务应该比这个复杂得多，例如进行机票预定等，这里演示目的。下面将创建一个客户端，在客户端中将调用这个 Web 服务。

创建一个名为 WebClientDemo 的 Web 项目。在“项目”窗口中，右键单击项目节点，选择弹出菜单中的“新建”→“文件/文件夹”命令，打开“新建文件”对话框。在“类别”列表中选择“Web 服务”节点，在“文件类型”列表中选择“Web 服务客户端”节点。单击“下一步”按钮，打开“新建 Web 服务客户端”对话框，如图 18-6 所示。

选中“项目”单选按钮，然后单击旁边的“浏览”按钮，打开“浏览 Web 服务”对话框，如图 18-7 所示。在 Web 服务树状视图中选择上一节创建的 MyWebService 服务。单击“确定”按钮返回“新建 Web 服务客户端”对话框。

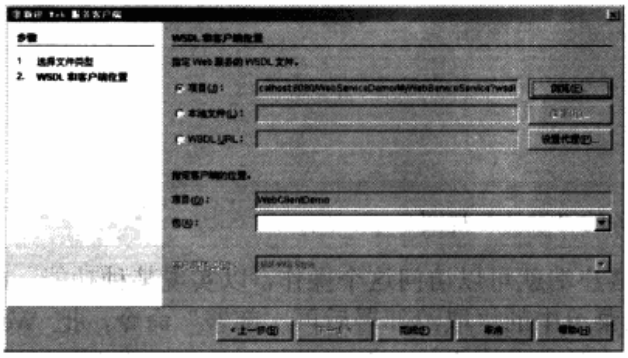


图 18-6 “新建 Web 服务客户端”对话框

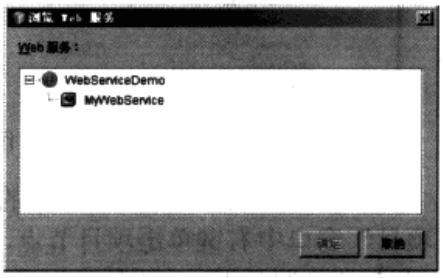


图 18-7 “浏览 Web 服务”对话框

说明：如果要从文件系统中选择 Web 服务的 WSDL 文件，那么需要选中“本地文件”单选按钮。如果要指定网络上的某个 WSDL 文件，则要选中“WSDL URL”单选按钮，然后在后面的文本框中输入 URL 地址。

在“包”下拉列表框中选择要使用的包名字，或者输入新包的名字。单击“完成”按钮，创建一个 Web 服务的客户端。此时，可以看到在“项目”窗口中，该项目节点下面的“Web 服务引用”节点下添加了对 MyWebService 的引用。

打开 index.jsp 文件，修改代码内容如下所示：

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>

    <h1>调用 Web 服务演示</h1>
    <form name="Form" action="CallService" method="POST">
        请输入第一个数：<input type="text" name="Num1" value="0" />
        <br>
        请输入第二个数：<input type="text" name="Num2" value="0" />
        <br>
        <input type="submit" value="提交" name="submit" />
    </form>
```

Download at http://www.pin5i.com/


```
    </body>
</html>
```

上述代码对应的运行界面如图 18-8 所示。其中，使用两个文本框输入要进行计算的数字。单击“提交”按钮后，将执行 CallService Servlet。

在项目中创建一个名字为 CallService 的 Servlet，该文件包含在 myweb 包中。在 processRequest()方法中单击鼠标右键，选择弹出菜单中的“Web 服务客户端资源”→“调用 Web 服务操作”命令，打开“选择要调用的操作”对话框，如图 18-9 所示。



图 18-8 Web 服务客户端的运行界面

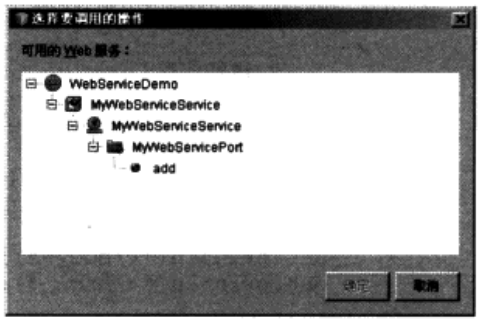


图 18-9 “选择要调用的操作”对话框

在“可用的 Web 服务”树状视图中选择 MyWebService 服务的 add 操作，单击“确定”按钮，自动添加如下的 Web 服务引用代码：

```
    @WebServiceRef(wsdlLocation = "http://localhost:8080/WebServiceDemo
/MyWebServiceService?wsdl")
    private MyWebServiceService service;
```

并在 processRequest 方法中添加如下的代码：

```
try { // Call Web Service Operation
    myservice.MyWebService port = service.getMyWebServicePort();
    // TODO initialize WS operation arguments here
    int num1 = 0;
    int num2 = 0;
    // TODO process result here
    int result = port.add(num1, num2);
    out.println("Result = "+result);
} catch (Exception ex) {
    // TODO handle custom exceptions here
}
```

其中，语句 int result = port.add(num1, num2)调用了 Web 服务的 add 操作，并把返回值保存在 result 中。下面修改 processRequest()方法中的代码，把用户输入的 num1 和 num2 作为参数传递给 add 操作。

```
    protected void processRequest(HttpServletRequest request, HttpServlet
Response response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet CallService</title>");
        out.println("</head>");
        out.println("<body>");

        //获得第一个数
        int num1 = 0;
        String num1Str = request.getParameter("Num1");
```

Download at http://www.pin5i.com/


```
        if (num1Str != null) {
            num1 = Integer.valueOf(num1Str);
        }

        //获得第二个数
        int num2 = 0;
        String num2Str = request.getParameter("Num2");
        if (num2Str != null) {
            num2 = Integer.valueOf(num2Str);
        }

        try { // Call Web Service Operation
            myservice.MyWebService port = service.getMyWebServicePort();

            //进行计算
            int result = port.add(num1, num2);
            out.println("<h1>Result = "+result+"</h1>");
        } catch (Exception ex) {
            ex.printStackTrace();
        }

        out.println("</body>");
        out.println("</html>");

        out.close();
    }
```

在图 18-5 所示的界面中输入数字后，单击“提交”按钮，将执行 CallService Servlet，运算结果将显示在如图 18-10 所示的界面中。

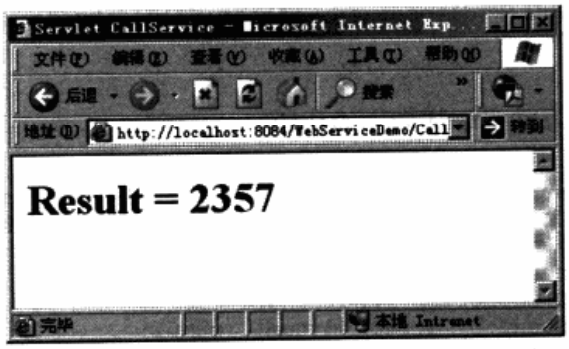


图 18-10 调用 Web 服务后的运算结果

18.2 创建 BPEL 模块项目

业务过程执行语言 BPEL（Business Process Execution Language）是一种基于 XML 的用来描写业务过程的编程语言，被视为 Web 服务的逻辑聚合器和协调器。

在 BPEL 流程中，一组合作伙伴 Web 服务组件可以按同步或异步方式进行协作，参与长时间运行的对话以及支持错误处理。NetBeans 集成开发环境提供了一个具有较高图形化程度的 BPEL 开发设计环境，包括编制、编辑、测试运行和调试 BPEL 流程的功能，以及部署运行环境。

BPEL 模块项目是一组源文件，其中包括 BPEL 文件、Web 服务接口文件（.wsdl）以及 XML 架构（.xsd）文件。在 BPEL 模块项目中，可以编制符合 WS-BPEL 2.0 语言规范的业务流程。

在 NetBeans 集成开发环境中，单击“文件”→“新建项目”命令，打开“新建项目”对话框。选择“类别”列表中的“SOA”节点，再选择“项目”列表中的“BPEL 模块”节点，如图 18-11 所示。

Download at http://www.pin5i.com/

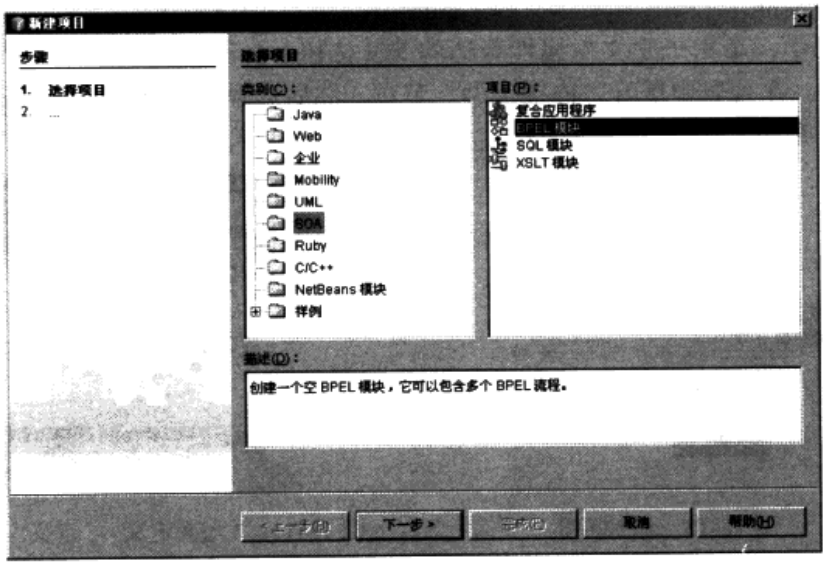


图 18-11 “新建项目”对话框

单击“下一步”按钮，打开“新建 BPEL 模块”对话框。在“项目名称”文本框中输入项目的名称，例如输入 MyBpelModule。选中“设置为主项目”复选框，单击“完成”按钮，即可创建一个 BPEL 模块。

18.3 创建 WSDL 文件

Web 服务描述语言 WSDL（Web Services Description Language）文件是基于 XML 语言的，用于描述 Web 服务。在 BPEL 模块项目中，可以使用 WSDL 文件为业务流程及其合作伙伴服务定义接口。

18.3.1 为 WSDL 创建 XML 架构文件

在创建 WSDL 文件前，可以先创建一个 XML 架构文件，用于定义一些数据类型。在“项目”窗口中，展开 BPEL 项目节点，右键单击“流程文件”节点，然后选择“新建”→“文件/文件夹”命令，打开“新建文件”对话框。

在“类别”列表中选择“XML”节点，然后在“文件类型”列表中选择“XML 架构”节点。然后单击“下一步”按钮，打开“新建 XML 架构”对话框。在“文件名”文本框中输入 XML 架构的名字 myXmlSchema，单击“完成”按钮。

在打开的可视化的 XML 架构编辑器中，单击“设计”按钮，打开“设计”视图。从组件面板中拖曳一个“复杂类型”组件，放置在“复杂类型”下面，如图 18-12 所示。

在图 18-12 所示的文本框中（双击该组件可以进入编辑模式）输入组件的名字，或者在属性编辑器中修改其“名称”属性，这里输入 myComplexType。

从组件面板中拖曳一个元素组件，放置在刚刚创建的复杂类型 myComplexType 下面，如图 18-13 所示。在图 18-13 所示的文本框中（双击该组件可以进入编辑模式）输入组件的名字，或者在属性编辑器中修改其“名称”属性，

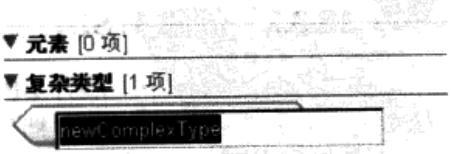


图 18-12 添加复杂类型组件

Download at http://www.pin5i.com/

这里输入 myElement。

在属性窗口中，单击“定义”属性旁边的 ... 按钮，打开“元素定义 – 定义”对话框，如图 18-14 所示。在列表中展开“内置类型”节点，然后选择“string”节点，指定元素的类型为字符串类型。单击“确定”按钮，关闭“元素定义 – 定义”对话框。

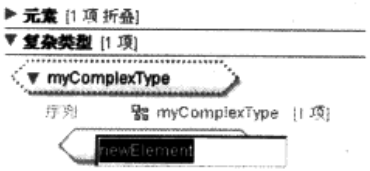


图 18-13 添加元素



图 18-14 “元素定义 – 定义”对话框

从组件面板中拖曳一个元素组件，放置在“元素”下面，创建一个全局元素，如图 18-15 所示。在图 18-15 所示的文本框中（双击该组件可以进入编辑模式）输入组件的名字，或者在属性编辑器中修改其“名称”属性，这里输入 typeA。

在属性窗口中，单击“定义”属性旁边的 ... 按钮，打开“元素定义 – 定义”对话框。在类型列表中，展开“复杂类型”节点，然后选择刚刚创建的 myComplexType 复杂类型节点。单击“确定”按钮，关闭“元素定义 – 定义”对话框。此时，会在 typeA 元素节点的旁边显示选定的 myComplexType 复杂类型，如图 18-16 所示。

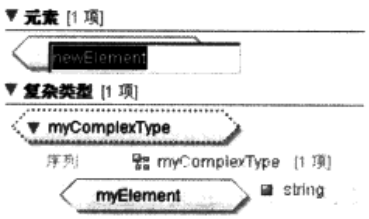


图 18-15 添加全局元素

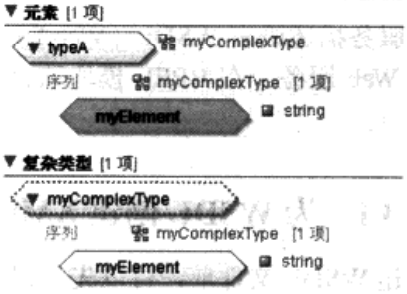


图 18-16 为全局元素指定复杂类型



说明：本小节的操作对于创建 WSDL 文件并不是必须的，用户如果只使用简单的内置数据类型，则可以直接创建 WSDL 文件。当用户需要定义自己的数据类型时，则要创建 XML 架构文件。



18.3.2 创建 WSDL 文件

在“项目”窗口中，展开 BPEL 项目节点。右键单击“流程文件”节点，然后选择弹出菜单中的“新建”→“WSDL 文档”命令。打开“新建 WSDL 文档”对话框，如图 18-17 所示。

在“文件名”文本框中输入文件的名称，不能输入项目中已存在的名称，这里输入 myWDSL。在“文件夹”文本框中输入文件创建的文件夹，这里输入 src。在“目标名称空间”文本框中输入 WSDL 文件的目标名称空间，目标名称空间将显示在根元素中，这里使用默认的名称空间。

Download at http://www.pin5i.com/

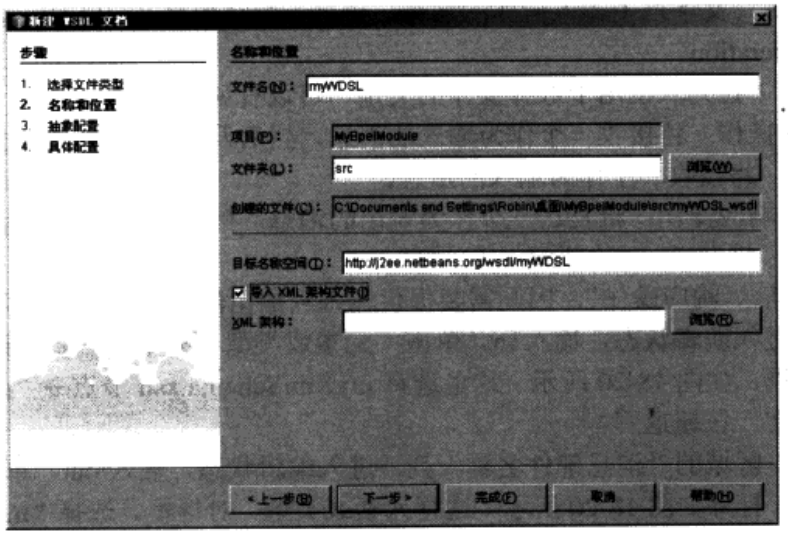


图 18-17 “新建 WSDL 文档”对话框

选中“导入 XML 架构文件”复选框，在 WSDL 文件中导入一个或多个 XML 架构文件。然后单击“XML 架构”文本框旁边的“浏览”按钮，打开“添加导入”对话框，如图 18-18 所示。这里选择上一节中创建的 myXmlSchema.xsd 文件，单击“确定”按钮。

在“新建 WSDL 文档”对话框中，单击“下一步”按钮，打开“抽象配置”对话框如图 18-19 所示。

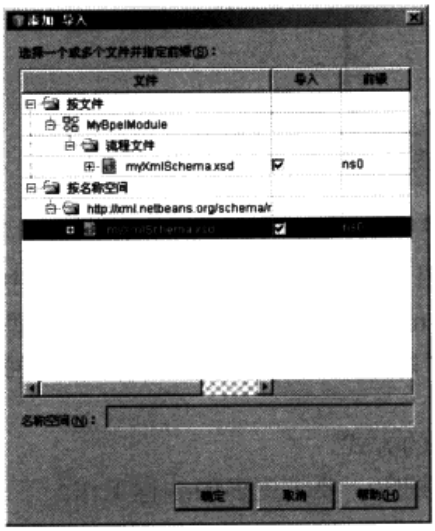


图 18-18 “添加导入”对话框

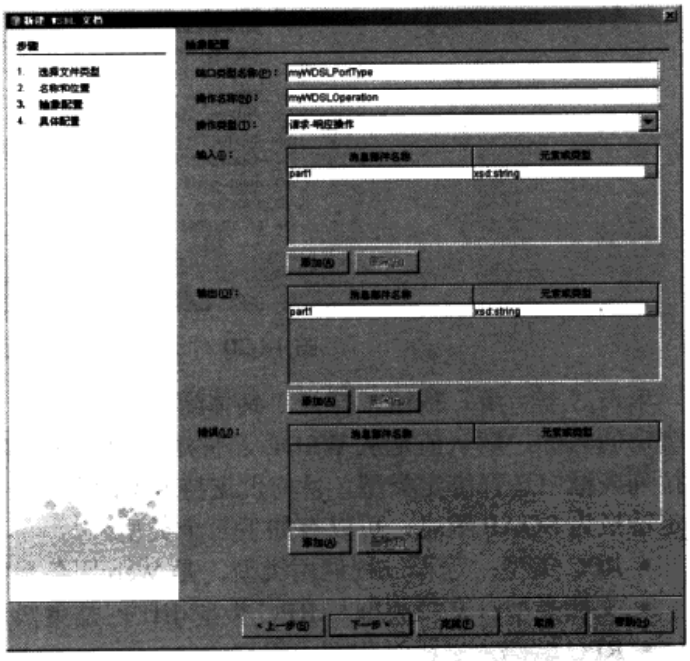


图 18-19 “抽象配置”对话框

在“端口类型名称”文本框中输入端口类型的名称，该名称默认值是从 WSDL 名称派生的，这里输入 myWSDLPortType。



注意：端口类型包含 Web 服务可执行的一个或多个相关操作。在“抽象配置”对话框中，只能创建一个操作。要创建其他操作，必须在创建 WSDL 文件后，在第 18.3.3 节中介绍的 WSDL 编辑器中执行创建操作。

Download at http://www.pin5i.com/

在“操作名称”文本框中输入操作的名称，该名称默认值是从 WSDL 名称派生的，这里输入 myWSDLOperation。

“操作类型”下拉列表框用于选择操作的类别，可以有如下的几个选择。

- 请求-响应操作：将接收一个作为输入的消息，然后发送一个作为输出的消息。
- 单向操作：将接收一个作为输入的消息。
- 逆向请求-响应操作：将发送一个作为输出的消息，然后接收一个作为输入的消息。
- 通知操作：将发送一个作为输出的消息。

这里选择“请求-响应操作”，因此需要提供输入和输出消息。双击“输入”区域的“消息部件名称”列，进入编辑状态，输入 in。单击“元素或类型”列中的 ... 按钮，打开“选择元素或类型”对话框，如图 18-20 所示。这里选择 myXmlSchema.xsd 节点→“元素”→“typeA”节点，单击“确定”按钮返回。

双击“输出”区域的“消息部件名称”列，进入编辑状态，输入 out。单击“元素或类型”列中的 ... 按钮，打开如图 18-20 所示“选择元素或类型”对话框。选择“myXmlSchema.xsd”节点→“元素”→“typeA”节点，单击“确定”按钮返回。

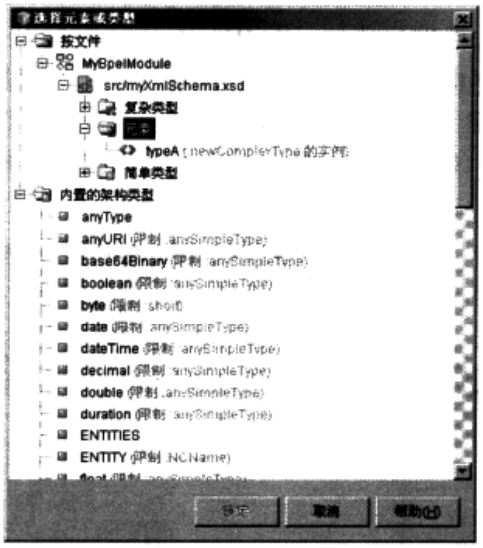


图 18-20 “选择元素或类型”对话框

单击“下一步”按钮，打开“具体配置”对话框，如图 18-21 所示。在“绑定名称”中输入绑定的名称，默认值是从 WSDL 名称派生的，这里输入 myWSDLBinding。在“绑定类型”下拉列表框中选择绑定类型，目前仅支持 SOAP 协议。在“绑定子类型”部分可以选择如何将绑定转换为 SOAP 消息，可以有如下三个类型。

- RPC 类型：远程过程调用类型，是 Web 服务最常用的方式。
- 文档类型：文档类型与 RPC 类型相比，是更低层的抽象，要求更多的编程工作。
- RPC 编码：远程过程调用编码。

这里，选择文档类型。在“服务名称”文本框中输入服务的名称，这里输入 myWSDLService。在“端口名称”文本框中输入端口的名称，这里输入 myWSDLPort。

单击“完成”按钮，将创建 WSDL 文件。关于 WSDL 文档编辑器的使用方法在下一节进行介绍。



说明：除了本节介绍的方法，还有其他两种创建 WSDL 文档的方法。在“新建文件”文本框中，选中“类别”列表中的“XML”节点，然后再选择“文件类型”列表中的

Download at http://www.pin5i.com/

“WSDL 文档”节点，可以创建 WSDL 文档。另外一种方法是在“新建文件”文本框中，选中“类别”列表中的“Web 服务”节点，再选择“文件类型”列表中的“WSDL”节点，也可以创建 WSDL 文档。

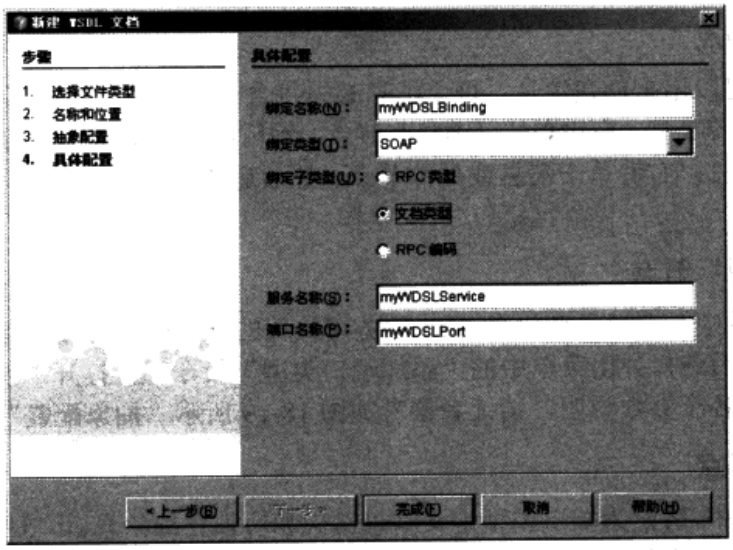


图 18-21 “具体配置”对话框

18.3.3 可视化的 WSDL 编辑器

WSDL 编辑器如图 18-22 所示，里面包括类型、导入、消息、端口类型、绑定、服务等几个部分。

1.“类型”节点

“类型”节点用于配置 WSDL 数据类型。右键单击“类型”节点，然后选择弹出菜单中的“添加”→“内联架构”命令，将显示内联架构节点。

右键单击“类型”节点，然后选择弹出菜单中的“添加”→“导入架构”命令，将打开“添加导入”对话框，如图 18-23 所示。在该对话框中可以按照文件或者名称空间的方式导入 XML 架构。

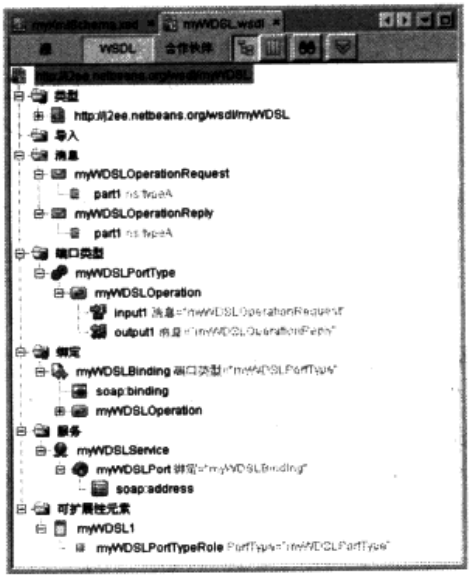


图 18-22 WSDL 编辑器

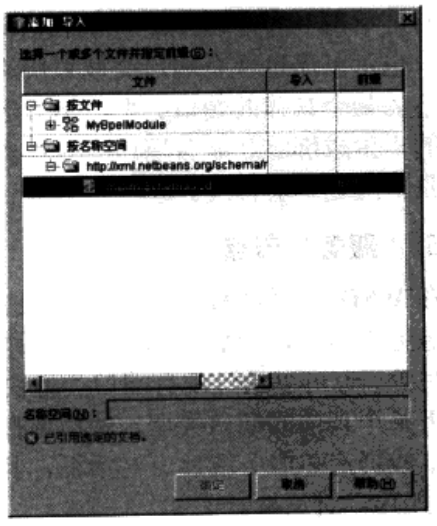


图 18-23 “添加导入”对话框

Download at http://www.pin5i.com/

2. “导入”节点

“导入”节点用于导入其他的 WSDL 文件。在“导入”节点上单击右键，选择弹出菜单中的“添加 导入 WSDL”命令，将打开“导入定制器”对话框，查找定位 WSDL 文件。

3. “消息”节点

“消息”节点可以用来创建消息。每个消息必须包含一个或多个逻辑部件。对于每个部件，必须指定名称和内容类型。右键单击“消息”节点，然后选择弹出菜单中的“添加消息”命令，将创建一个新的消息。右键单击消息或部件节点，然后选择“删除”命令，将删除选定的消息或部件。

4. “端口类型”节点

在 WSDL 文件中，端口类型包含 Web 服务可执行的一个或多个相关操作。右键单击“端口类型”节点，然后选择弹出菜单中的“添加端口类型”命令，将打开“创建新端口类型”对话框。该对话框的各个参数说明，请读者参考如图 18-19 所示“抽象配置”对话框。

5. “绑定”节点

在WSDL文件中，绑定为端口类型定义消息格式和协议详细信息。可以绑定一个遵循SOAP 1.1 协议的绑定。该绑定包含一个 SOAP 绑定节点以及一个或多个操作，如图 18-24 所示。每个操作可以包含以下元素。

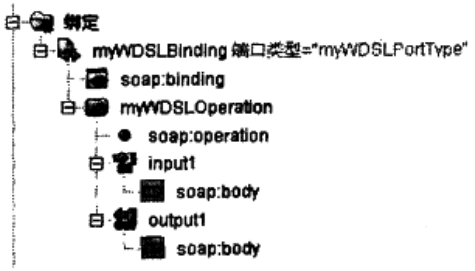


图 18-24 “绑定”节点

- soap:body 元素：提供 SOAP 正文元素的信息。
- soap:header 和 soap:headerfault 元素：提供 SOAP 头标元素的信息。
- soap:fault 元素：提供 SOAP 错误详细信息元素的信息。

右键单击“绑定”节点，然后选择弹出菜单中的“添加绑定”命令，将创建一个绑定节点。另外，也可以通过现有端口类型来快速创建绑定，具体方法介绍如下。右键单击“端口类型”节点下的某个具体的端口类型，然后选择弹出菜单中的“添加”→“绑定和服务端口”命令。打开的“生成绑定和服务端口”对话框中，各个参数的使用方法，请读者参考如图 18-21 所示的“具体配置”对话框。

6. “服务”节点

在 WSDL 文件中，服务指示要使用的绑定以及客户端可以访问 Web 服务的位置。右键单击“服务”节点，然后选择弹出菜单中的“添加服务”命令，将添加一个服务。

另外，也可以通过现有端口类型快速创建服务，具体方法介绍如下。右键单击“端口类型”节点下某个具体的端口类型，然后选择弹出菜单中的“添加”→“绑定和服务端口”命令。打开“生成绑定和服务端口”对话框，其中各个参数的使用方法，请读者参考如图 18-21 所示的“具体配置”对话框。

Download at http://www.pin5i.com/

18.4 开发 BPEL 流程

18.4.1 创建 BPEL 流程

在“项目”窗口中，展开 BPEL 模块项目节点，右键单击“流程文件”节点，然后选择“新建”→“文件/文件夹”命令，打开“新建文件”对话框。

在“类别”列表中选择“面向服务的体系结构”节点，然后在“文件类型”列表中选择“BPEL 流程”节点。单击“下一步”按钮，打开“新建 BPEL 流程”对话框，如图 18-25 所示。

在“文件名”文本框中输入 BPEL 流程文件的名字，这里输入 myProcess。“目标名称空间”文本框中用于输入名称空间，这里使用默认值。单击“完成”按钮，将创建一个 BPEL 流程。

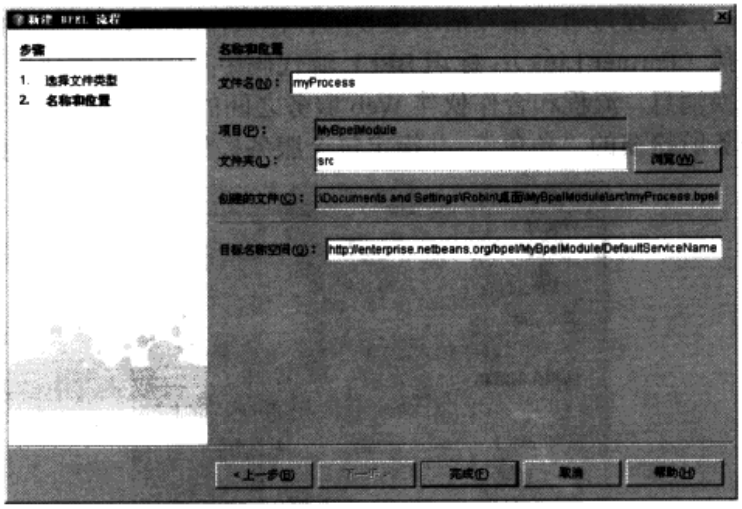


图 18-25 “新建 BPEL 流程”对话框

18.4.2 BPEL 流程编辑器

创建的 BPEL 流程会自动在可视化的 BPEL 编辑器中打开，如图 18-26 所示。也可以在“项目”窗口中，通过双击“.bpel”文件打开。

在“设计”视图中，可以添加、编辑和删除图元素。“设计”视图中构造的图将自动生成符合 BPEL 2.0 语言规范的 BPEL 源代码。在“源”视图中，可以编辑业务流程的源代码。在保存源文件中的修改内容后，BPEL 图也会相应地与源文件保持同步。

当处理较大或者较小的图时，可以使用缩放功能来缩小或放大图的尺寸，以便查看缩略图的更多内容或放大图的更详细内容。移动工具栏上的滑块，可以放大或者缩小图形。

- 单击“适当地调整图大小”按钮：可以在窗口中查看整个图。

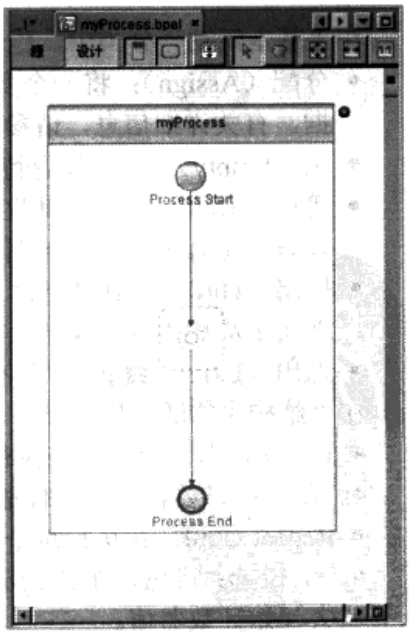


图 18-26 可视化的 BPEL 编辑器

Download at http://www.pin5i.com/

- 单击“适当地调整宽度大小”按钮：可以查看完整宽度的图。
- 100%按钮：以正常比例显示流程图。



18.4.3 BPEL 流程元素

当打开 BPEL 流程编辑器的“设计”视图时，会显示如图 18-27 所示的组件面板。BPEL 流程编辑器的“设计”视图支持拖曳操作，可以把组件面板中的组件添加到“设计”视图中。

在“Web 服务”分类中，包含如下几个元素。

- 调用(Invoke)：允许业务流程向合作伙伴 Web 服务发送消息。在合作伙伴提供的 portType 上调用单向操作或请求-响应操作。该操作是在合作伙伴的 WSDL 文件中定义的。
- 接收（Receive）：流程等待来自合作伙伴服务的匹配消息到达。流程只有收到来自合作伙伴服务的匹配消息时才会继续或结束。
- 回复（Reply）：流程为通过接收活动接收的合作伙伴服务消息发送回复消息。
- 合作伙伴链接（Partner Link）：标识 BPEL 源代码中列出的合作伙伴 Web 服务，它将与业务流程交换消息。流程和合作伙伴 Web 服务之间的关系是通过以下方法进行描述的。定义每个服务所扮演的“角色”，并指定每个服务提供的用于接收消息的 portType。

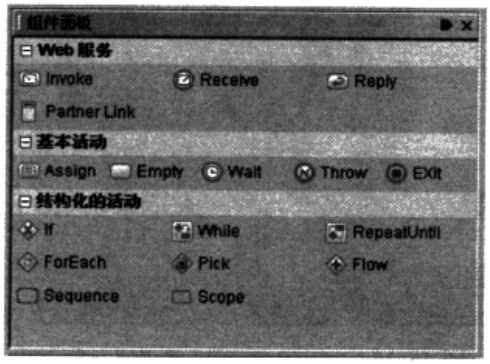


图 18-27 BPEL 流程元素

在“基本活动”分类中，包含如下几个元素。

- 分配（Assign）：将一个变量的值复制到另一个变量，或者使用表达式为变量分配新数据。有关详细信息，请参见关于 BPEL 映射器。
- 空（Empty）：表示没有特定活动，因而在需要捕获错误和禁止错误时非常有用。
- 等待（Wait）：指示流程在指定的时间内等待，或者直至某一特定时间后结束等待。必须只满足其中的一个到期条件。
- 抛出（Throw）：在业务流程内生成错误，或者在业务流程内抛出异常。通过它可以显式地发出流程内部出错的信号。
- 退出（Exit）：终止执行业务流程实例。

在“结构化的活动”分类中，包含如下几个元素。

- If：在根据一组选项来执行某分支活动时，将指定条件流程行为。
- While：只要指定的条件成立，就会重复执行某个活动。
- Repeat Until：在执行每个迭代后，只要特定条件成立，就会反复执行一个或多个活动。
- For Each：将包含在 Scope 活动内的活动重复执行 N+1 次，其中 N 等于最终计数器值减去起始计数器值。
- Flow：指定一个或多个要并发执行的活动（依次指定）。
- Sequence：定义一组要按列出的顺序执行的活动。

Download at http://www.pin5i.com/

- Pick：等待发生一个定义的事件，然后执行与发生的事件相关联的活动。通常，这些事件的发生是互斥的。如果发生了多个事件，则会执行先发生事件的活动。
- Scope：使用活动自身关联的局部变量、错误处理程序、合作伙伴链接以及关联集来定义嵌套活动。



18.4.4 简单的 BPEL 流程实例

1. 添加一个合作伙伴

在“项目”窗口中，展开前面创建的 MyBpelModule 项目下的“流程文件”节点。选择“myWDSL.wsdl”节点，将其从“项目”窗口拖到“设计”视图中。将打开“新建 Partner Link”对话框，如图 18-28 所示。

该对话框用于添加合作伙伴链接。在 BPEL 语言中，合作伙伴链接是对 BPEL 流程交互的外部 Web 服务的引用。合作伙伴链接活动用于定义每个合作伙伴在对话消息交换中所扮演的角色。依靠其对合作伙伴链接类型的引用，合作伙伴链接活动间接地指示每个合作伙伴实现的端口类型。这里，接受默认值，然后单击“确定”按钮，添加一个合作伙伴，如图 18-29 所示。

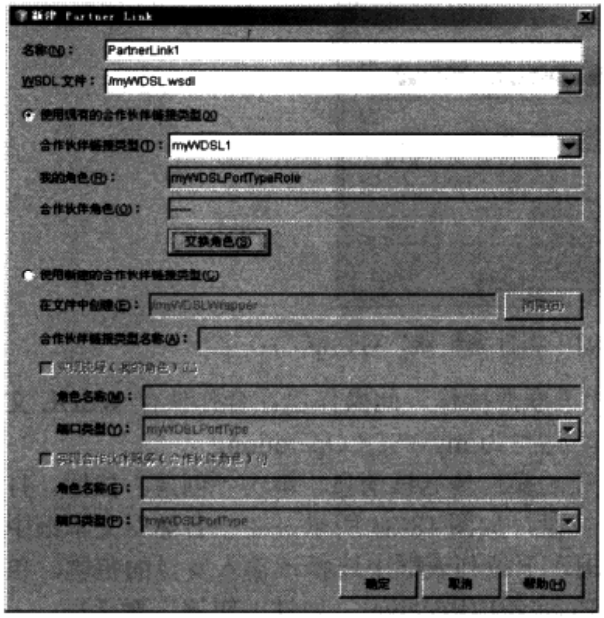


图 18-28 “新建 Partner Link”对话框

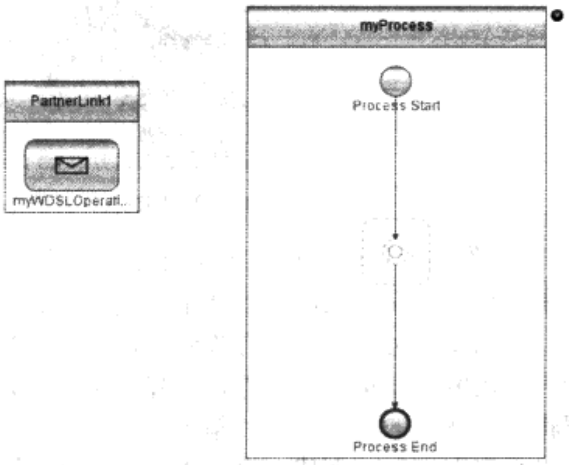


图 18-29 添加一个合作伙伴

2. 添加接收活动

在组件面板的“Web 服务”类别中，选择 Empty 图标，然后将所选的活动拖到设计区域中 Process Start 活动和 Process End 活动之间，如图 18-30 所示。再选择 Receive 图标，然后将所选的活动拖到设计区域中 Process Start 活动和 Empty 节点之间。释放鼠标后，显示界面如图 18-31 所示。

双击“Receive1”活动，打开“Receive1 [Receive] - 属性编辑器”对话框，如图 18-32 所示。“名称”文本框内为接收活动的名字。“合作伙伴链接”下拉列表框用于选择应发送入站消息的合作伙伴链接，这里选择 PartnerLink1。



说明：将消息流拖到图中，也将合作伙伴链接与接收活动相关联。

Download at http://www.pin5i.com/

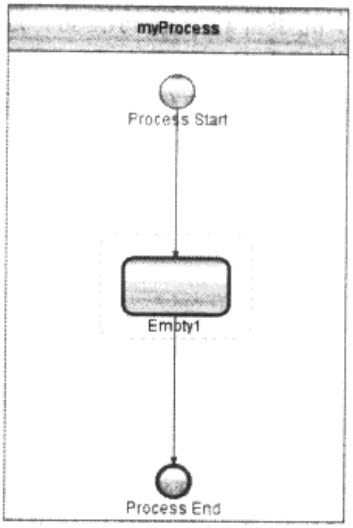


图 18-30 添加接收活动

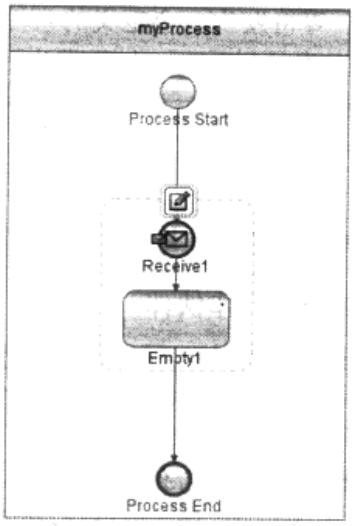


图 18-31 已添加的接收活动

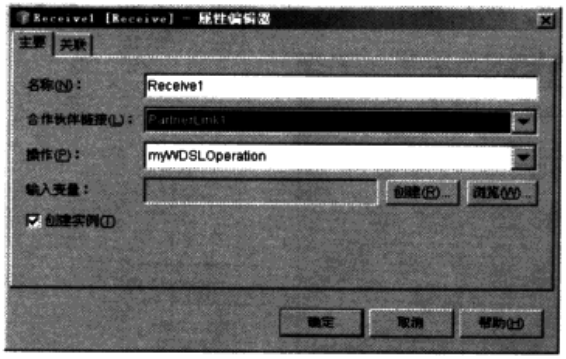


图 18-32 “Receive1 [Receive] - 属性编辑器”对话框

“操作”下拉列表框用于定义要与接收元素相关联的操作。此操作已在合作伙伴 WSDL 文件中进行了定义，因此可以从下拉列表中选择它，这里为 myWSDLOperation。

输入变量可以用来保存收到的消息，并为后续的操作传送该消息。单击“创建”按钮，打开“新建输入变量”对话框，创建一个新的输入变量，如图 18-33 所示。在“名称”文本框中编辑输入变量的名字，这里输入 inVar。在“范围”下拉列表框中选择该输入变量的范围。单击“确定”按钮，创建一个输入变量，并返回“Receive1 [Receive] - 属性编辑器”对话框。

最后，选中“创建实例”复选框，表示此接收元素会启动业务流程。单击“确定”按钮，返回 BPEL 编辑器。

3. 添加回复活动

在组件面板的“Web 服务”类别中，选择 Reply 图标，然后将所选的活动拖到设计区域中 Process End 活动和空活动之间。释放鼠标后，显示界面如图 18-34 所示。

双击“Reply1”活动，打开“Reply1 [Reply] - 属性编辑器”对话框，如图 18-35 所示。“名称”文本框内为回复活动的名字。“合作伙伴链接”下拉列表框用于选择应发送入站消息的合作伙伴链接，这里选择 PartnerLink1。

“操作”下拉列表框用于定义要与接收元素相关联的操作。此操作已在合作伙伴 WSDL 文件中进行了定义，因此可以从下拉列表中选择它，这里为 myWSDLOperation。



说明：将消息流拖到图中，也将合作伙伴链接与回复活动相关联。

Download at http://www.pin5i.com/

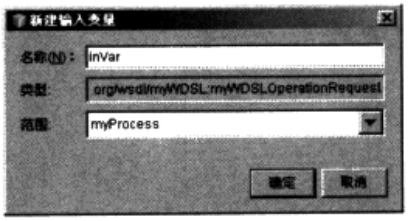


图 18-33 “新建输入变量”对话框

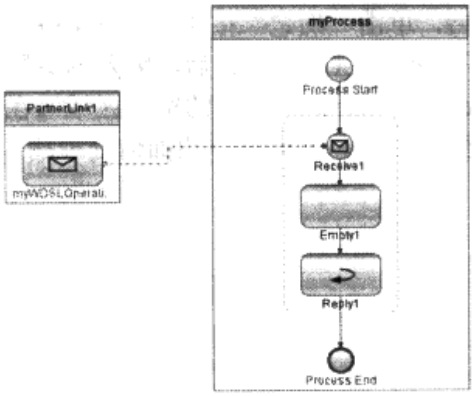


图 18-34 添加的回复活动

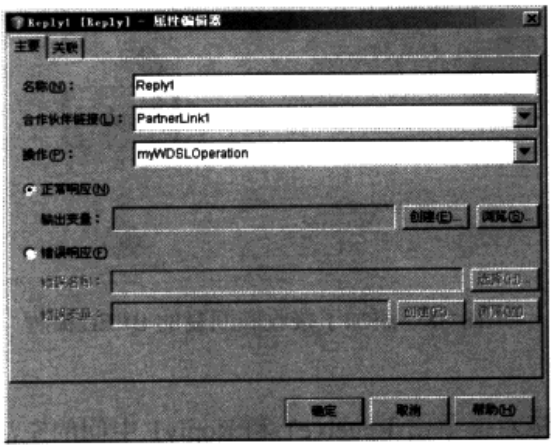


图 18-35 “Reply1 [Reply] - 属性编辑器”对话框

回复的响应类型有两种：正常响应和错误响应。其中，错误响应可将回复元素配置为发送错误消息。正常响应可将回复元素配置为发送正常响应类型的消息。

这里选择“正常响应”单选按钮后，单击“创建”按钮，将打开如图 18-36 所示的“新建输出变量”对话框，创建新的输出变量。

在“名称”文本框中编辑输入变量的名字，这里输入 outVar。在“范围”下拉列表框中选择该输入变量的范围。单击“确定”按钮，创建一个输出变量，并返回“Reply1 [Reply] - 属性编辑器”对话框。如果要选择已经存在的输出变量，可以单击“浏览”按钮，打开“输出变量选择器”对话框选择输出变量，如图 18-37 所示。

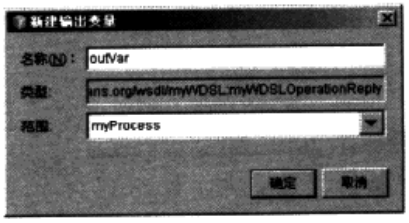


图 18-36 “新建输出变量”对话框

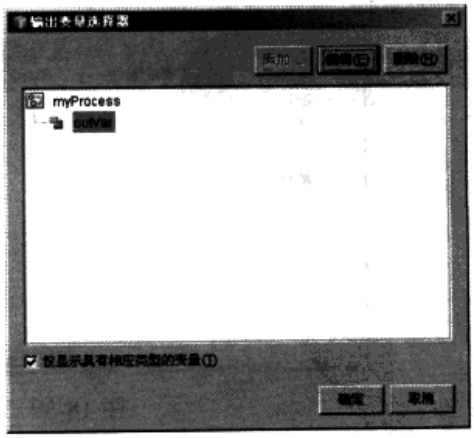


图 18-37 “输出变量选择器”对话框

Download at http://www.pin5i.com/

最后，单击“确定”按钮，返回 BPEL 编辑器，如图 18-38 所示。当客户端调用 Web 服务的操作 myWSDLOperation 后，输入的参数就会传递给 Receive1，进行 BPEL 业务流程进行处理(Receive1 和 Reply1 中间的空白区域)。在处理完毕后，由 Reply1 再传递给 myWSDLOperation 并返回到客户端。下面将介绍如何处理输入 Receive 活动，即如何分配活动。

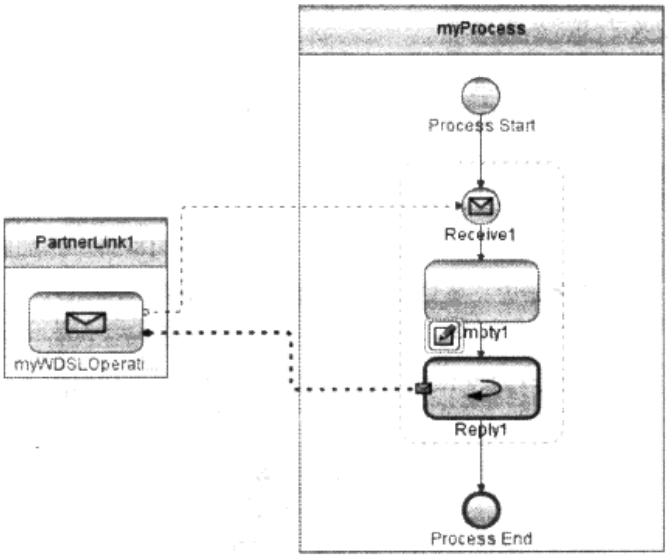


图 18-38 添加了接收和回复的 BPEL 流程

4. 添加分配活动

在设计区域中，用鼠标右键单击 Receive1 和 Reply1 中间的空活动，然后选择弹出菜单中的“删除”命令，删除这个空操作。在组件面板的“基本活动”类别中，选择 Assign 图标，然后将其拖到设计区域中 Receive1 和 Reply1 活动之间。释放鼠标后，将放置一个名为 Assign1 的分配活动。单击 Assign1 活动，打开“BPEL 映射器”窗口，如图 18-39 所示。

BPEL 映射器提供了处理和定向业务流程数据的框架。最左边的是源树窗格，其中包含了一个树组件，用于提供对业务流程的源数据变量的访问。最右边的是目标树窗格，其树组件的内容与要映射的业务流程元素有关：

- 对于分配活动，右窗格包含与左窗格相同的树组件；
- 对于 If 活动、ElseIf 元素、ForEach 活动、RepeatUntil 活动以及 While 活动，右窗格都包含“结果”节点；
- 对于等待活动和活动警报事件，右窗格包含“截止期限”或“持续时间”节点。

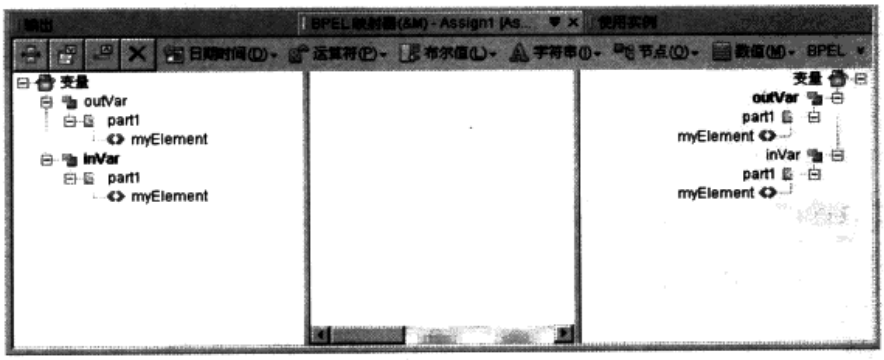


图 18-39 “BPEL 映射器”窗口

中间的部分为映射窗格，用于创建 BPEL 映射的画布。根据业务流程元素的不同，在映射

Download at http://www.pin5i.com/

窗格中执行的操作也不同。例如对于分配活动，则可以定义一个或多个复制赋值；对于 If 活动、If 活动内的 ElseIf 元素、ForEach 活动、RepeatUntil 活动、While 活动等都可以定义条件；对于等待活动和活动警报事件，则可以指定截止期限或持续时间。

BPEL 映射器的工具栏提供了所有用来创建 BPEL 映射的 BPEL 映射器 XPath 函数。还可以通过加入包含 XPath 函数的谓词来改进或扩展 BPEL 映射。当在菜单栏中选择函数时，将在映射窗格中显示一个函数框。如果该函数接受任何参数，则函数框的左侧将为每个参数对应一个连接器。如果参数是可选的，则会在参数名称后面显示一个问号。并且函数框的右侧也会为结果对应一个连接器。

这里，把左边窗格输入变量 inVar 中的 myElement 元素拖曳到右边窗格输出变量 outVar 中的 myElement 元素上，如图 18-40 所示。创建一个简单的映射，即把输入值直接传递给输出变量。

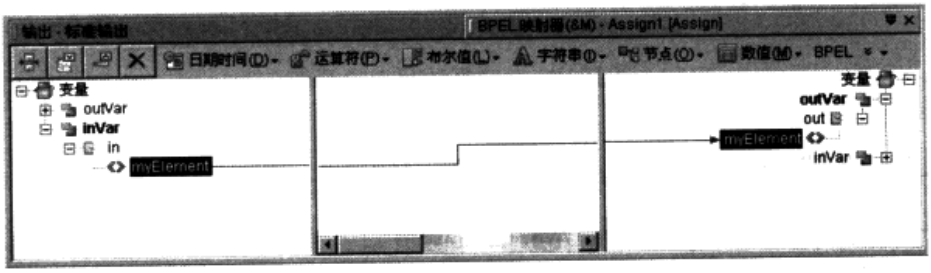


图 18-40 创建简单的映射

至此，完成了一个从输入→处理输入→输出的简单业务流程。在第 18.5 节中将在 SOA 应用程序中使用这个流程。

18.5 开发 SOA 复合应用程序

本节将创建一个简单的 SOA 复合应用程序，演示如何进行 SOA 复合应用程序的开发。在这个程序中使用到了 XML 框架、WSDL 文档、BPEL 模块。在 SOA 部署到服务器以后，将创建以上 Web 客户端程序进行测试。

18.5.1 创建 SOA 复合应用程序

在 NetBeans 集成开发环境中，选择“文件”→“新建项目”命令，打开“新建项目”对话框。在“类别”列表中，选择“面向服务的体系结构”节点。在“项目”列表中，选择“复合应用程序”节点。单击“下一步”按钮，打开“新建复合应用程序”对话框。

在“项目名称”字段中，复合应用程序的名字，这里输入 MyCompositeApp。单击“完成”按钮，创建一个空白的复合应用程序。

在“项目”窗口中，右键单击复合应用程序节点，选择弹出菜单中的“添加 JBI 模块”命令，打开“选择项目”对话框，如图 18-41 所示。

在选择 BPEL 模块前，首先要确认已经生成了 BPEL 模块。这样当选择了 BPEL 项目后，会在“项目 JAR 文件”列表中看到.jar 文件。这里，选择前面创建的 MyBpelModule 项目，单击“添加项目 JAR 文件”按钮。此时，会在“项目”窗口中，BPEL 项目的“JBI 模块”子节点下面看到已经添加了一个*.jar 文件。

生成项目，然后在“项目”窗口中右键单击项目节点，选择弹出菜单中的“部署”命令。把复合应用程序部署到服务器端。

Download at http://www.pin5i.com/

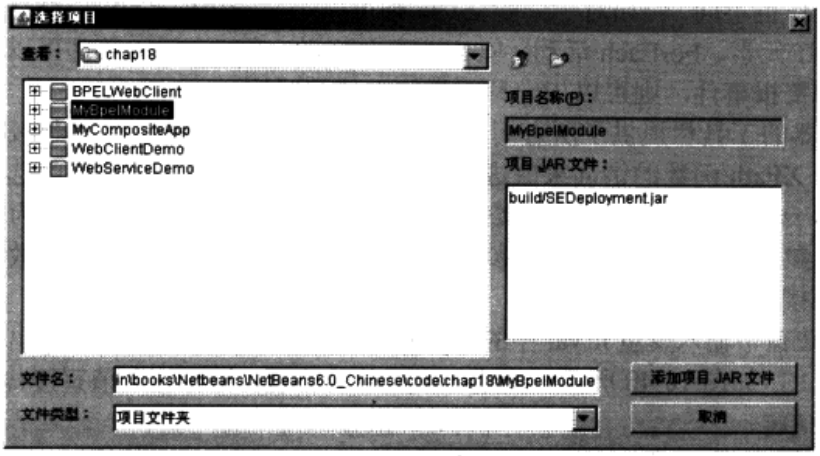


图 18-41 “选择项目”对话框

18.5.2 创建 Web 客户端测试 SOA 程序

创建一个标准 Web 应用程序 BPELWebClient，并设置为主项目。按照第 18.1.2 节中的方法创建一个 Web 服务客户端。从本地文件系统中添加 myWSDL.wsdl 文件，具体方法请参考第 18.1.3 节。下面编写提交用户输入的界面，在 index.jsp 文件中修改代码如下所示：

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>SOA 程序测试客户端－输入界面</title>
    </head>
    <body>

        <h1>SOA 程序测试客户端－输入界面</h1>
        <form action="SimpleSOAClient.jsp">
            <input type="text" name="input" value=""/>
            <input type="submit" value="提交" name="submit" />
        </form>
    </body>
</html>
```

再创建一个名为 SimpleSOAClient.jsp 的文件。在文件中单击右键，选择弹出菜单中的“Web 服务客户端资源”→“调用 Web 服务操作”命令，打开“选择要调用的操作”对话框。在该对话框中，选择前面创建的 myWDSLOperation 操作，自动创建如下的代码：

```
<%-- start web service invocation --%><hr/>
<%
    try {
        org.netbeans.j2ee.wsdl.mywdsl.MyWDSLService service = new
org.netbeans.j2ee.wsdl.mywdsl.MyWDSLService();
            org.netbeans.j2ee.wsdl.mywdsl.MyWDSLPortType port =
service.getMyWDSLPort();
    // TODO initialize WS operation arguments here
    org.netbeans.xml.schema.myxmlschema.MyComplexType in = null;
    // TODO process result here
    org.netbeans.xml.schema.myxmlschema.MyComplexType result =
port.myWDSLOperation(in);
        out.println("Result = " + result);
    } catch (Exception ex) {
    // TODO handle custom exceptions here
```

Download at http://www.pin5i.com/


```
        }
    %>
    <%-- end web service invocation --%><hr/>
```

修改调用 Web 服务的代码，把用户的输入赋值给输入标量 in。此时的 SimpleSOAClient.jsp 文件内容如下所示：

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>SOA 程序测试客户端一输出界面</title>
    </head>
    <body>

        <h1>SOA 程序测试客户端一输出界面</h1>

        <%-- start web service invocation --%><hr/>
        <%

            String input = request.getParameter("input");
            if (input == null) {
                input = "";
            }
            try {
                org.netbeans.j2ee.wsdl.mywdsl.MyWDSLService service = new
org.netbeans.j2ee.wsdl.mywdsl.MyWDSLService();
                org.netbeans.j2ee.wsdl.mywdsl.MyWDSLPortType port =
service.getMyWDSLPort();
                // TODO initialize WS operation arguments here
                org.netbeans.xml.schema.myxmlschema.MyComplexType in =
                        new org.netbeans.xml.schema.myxmlschema.MyComplexType();
                // TODO process result here

                //把用户输入赋值给输入变量
                in.setMyElement(input);

                org.netbeans.xml.schema.myxmlschema.MyComplexType result =
port.myWDSLOperation(in);
                out.println("Result = " + result.getMyElement());
            } catch (Exception ex) {
                // TODO handle custom exceptions here
                out.println(ex.toString());
            }
        %>
        <%-- end web service invocation --%><hr/>

        <form action="index.jsp">
            <input type="submit" value="返回主页面"/>
        </form>
    </body>
</html>
```

运行项目，显示如图 18-42 所示的用户输入界面。在输入“Hello”后，单击“提交”按钮，将显示如图 18-43 所示的输出界面。

18.6 小结

本章主要介绍基于 Web 服务的开发，包括创建简单的 Web 服务，以及使用 Web 服务的客户端。对于 SOA 应用程序，因为其内容比较繁多，因此主要通过一个实例进行介绍。在这个

Download at http://www.pin5i.com/

SOA 程序中，介绍了开发 WSDL 文档、BPEL 流程等技术。通过本章的学习，读者应该掌握以下基本内容：

- 如何创建一个 Web 服务
- 如何添加 Web 操作
- 如何在客户端中使用 Web 服务
- 如何创建 WSDL 文档，以及如何使用 WSDL 编辑器
- 如何创建一个 BPEL 模块，以及如何进行流程设计
- 如何创建 SOA 程序并添加 BPEL 模块

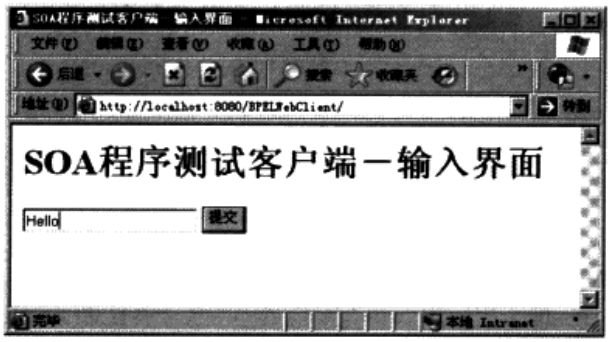


图 18-42 SOA 客户端输入界面

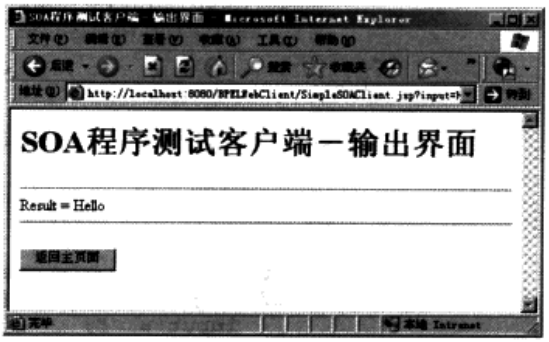


图 18-43 SOA 客户端输出界面

Download at http://www.pin5i.com/

PART 3

第三篇　NetBeans 手机开发

第 19 章　创建和运行 Java ME 应用程序
第 20 章　Java ME 程序的可视化开发
第 21 章　Java ME 开发高级技术

Download at http://www.pin5i.com/

第 19 章

创建和运行 Java ME 应用程序

Java ME 平台是针对小型设备（如寻呼机、移动电话和机顶盒）开发的一组技术和规范。Java ME 定义了许多专门针对电子消费设备和嵌入式设备的 API。在 NetBeans 集成开发环境中通过安装 Mobility Pack 软件包，可以快速开发出 Java ME 应用程序。

本章主要介绍 Mobility Pack 软件包，讲解摩托罗拉 Java ME 开发包的安装和使用，简单 Java ME 应用程序的创建和运行，以及针对 Java ME 应用程序的各种配置。

本章主要内容

- Mobility Pack 软件包的简要介绍
- 摩托罗拉的 Java ME 开发包的安装和在 NetBeans 中的集成
- 简单 Java ME 应用程序的创建
- Java ME 应用程序的配置

本章亮点

- 在 NetBeans 中集成摩托罗拉 Java ME 开发包
- 创建简单的 Java ME 应用程序
- 使用项目配置
- 部署 Java ME 应用程序

19.1 Mobility Pack 软件包简介

Mobility Pack 是用于开发基于 Java ME 平台应用程序的 NetBeans 软件包。Mobility Pack 提供了 Java ME 应用程序的可视化开发环境，大大提高了开发的效率并简化了开发的难度。Mobility Pack 主要有以下几个特点。

- 支持 Connected Limited Device Configuration（CLDC）1.0 和 CLDC 1.1、Mobile Information Device Profile（MIDP）1.0 及 2.0、2.1 版。
- 可视化设计器——Visual Mobile Designer（VMD）。拖放组件到画布上，轻松实现应用程序的流设计和屏幕的设计。
- Mobile Game Builder——专门为 MIDP 2.0 Game API 设计的可视编辑器。在 Game Builder 中可以使用工具创建 Scene、Tiled Layer 以及 Sprite。
- 支持定制组件。Mobility Pack 提供非常实用的定制组件，如等待屏幕、登录屏幕、闪现屏幕等。用户也可以定制自己的组件，而这一切实现起来非常便捷。
- 提供解决设备分裂问题的方案。在项目中通过创建创建多种设备的配置来解决设备分裂问题，这些配置都可以通过向导来完成。
- 访问 Web 服务。如果设备支持 JSR-172，就可以使用 MIDlets 通过使用无线连接桥来访问 Web 服务（或其他服务器端数据）。
- 支持 SVG 图像。可以 Mobility Pack 中预览并编辑 SVG 图像。同时 Mobility Pack 还提

Download at http://www.pin5i.com/

供了支持 SVG 图像的定制组件，例如 SVG 等待屏幕、SVG 菜单等。

- 其他集成工具。Mobility Pack 还提供了其他很多实用的工具来辅助开发。如 MIDlet 签名、推注册模拟、基于 SMS 和 CBS 信息的 WMA 模拟以及多种部署方法等。

在使用 NetBeans 开发 Java ME 应用程序前，必须确保系统中已经安装带有 Mobility Pack 的 NetBeans IDE。NetBeans 的 Mobility 软件包可以在 www.netbeans.org 进行下载。

在下载时请选择“Mobility”版本或“All”版本，单击相应版本下面的“下载”按钮即可。

19.2 安装 Java ME 开发包

在实际开发中，往往要针对某一个或者几个机型的手机进行开发，因此需要用到专门为这些手机设计的 Java ME 开发包。本节将以 Motorola Java ME SDK v6.3 for Motorola OS Products 开发包为例介绍如何安装其他手机厂商的 Java ME 开发包，以及如何把开发包集成到 NetBeans 集成开发环境中。

19.2.1 安装摩托罗拉的 Java ME 开发包

Motorola Java ME SDK v6.3 for Motorola OS Products 开发包可以在摩托罗拉的 Java ME 官方支持网站 developer.motorola.com 下载。摩托罗拉的 Java ME 开发包是免费下载的，但是下载前需要首先在该网站注册一个用户。

Motorola Java ME SDK v6.3 for Motorola OS Products 开发包中包含了如下机型的仿真器，用户可以选择需要的仿真器进行程序的开发。

- Emulator A.1 系列仿真器：A630、A845、C380、C381p、C381/C386、C384/C385/C390/C391、C650/C651、C698p、E375、E398、E550/V535/V560、T725、V180/V185/V186/V188、V220/V220i/V226、RAZR V3(CLDC 1.0)/RAZR V3b、V300/V400/V500、V330/V547/V551/V555、V303p/V400p、V540/V545/V550、V600、V600i、V620、V635、V80。
- Emulator A.3 系列仿真器：C975、C980、E770、E1000/E1000R、E1070、RAZR V3x、V975、V980。
- Emulator A.4 系列仿真器：L2/L6、PEBL U6、SLVR L7、RAZR V3、RAZR V3e/V3i、ROKR E1、V190/V191、V195/V196/V197、V230/V235、V360/V361、V365、V557/V557p。
- Emulator A.5 系列仿真器：MOTORAZR maxx V6、MOTORAZR V3xx、V1100。
- Emulator A.6 系列仿真器：MOTOKRZR K1、MOTORIZR Z3、MOTOSLVR L7i/L7e。

在安装之前请首先确认系统必须为 Windows 2000 或者 Windows XP，并且系统中已经安装并配置了 Java 2 SDK 1.4.1_01 或者以后的版本。

下载完成后，解压下载文件。双击 Mot_6.3.exe 文件进行安装，打开如图 19-1 所示的安装版本信息界面。该窗口消失后显示欢迎信息窗口，如图 19-2 所示。

单击 Next 按钮，打开版权声明窗口，如图 19-3 所示。单击 Yes 按钮接受版权声明后，打开选择安装路径的对话框，如图 19-4 所示。

用户可以单击 Browse 按钮选择其他的安装路径。单击 Next 按钮，在安装前进行最后的检查，如图 19-5 所示。确认安装信息无误后，单击 Next 按钮开始安装程序。安装完毕后显示如图 19-6 所示的完成对话框。

Download at http://www.pin5i.com/



图 19-1 安装版本信息界面

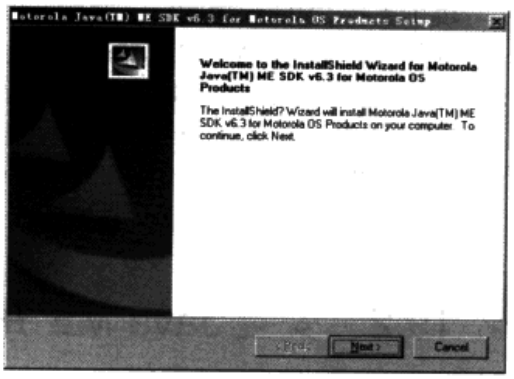


图 19-2 欢迎信息窗口

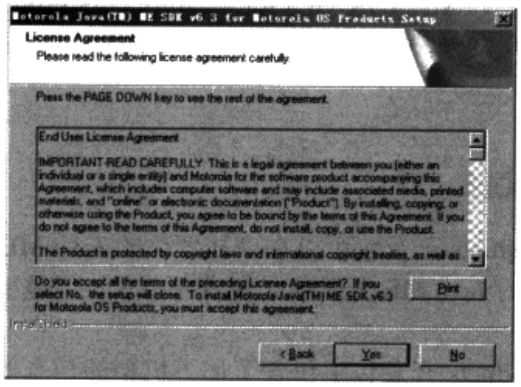


图 19-3 版权声明窗口

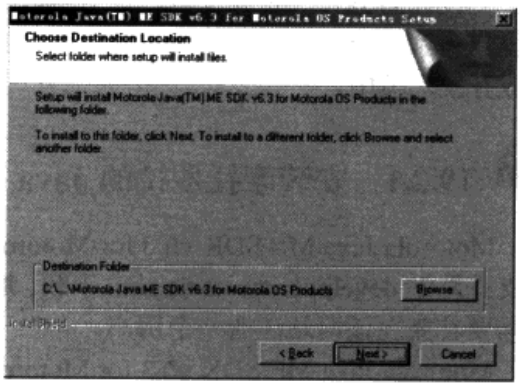


图 19-4 选择安装路径对话框

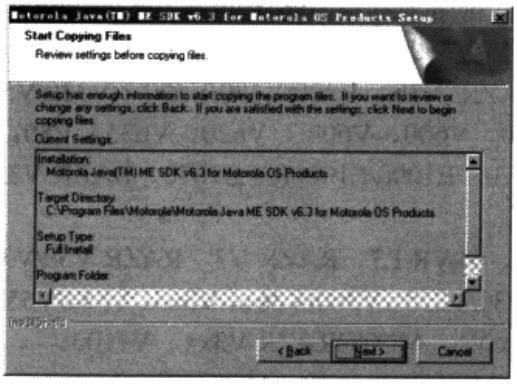


图 19-5 安装前进行最后的检查

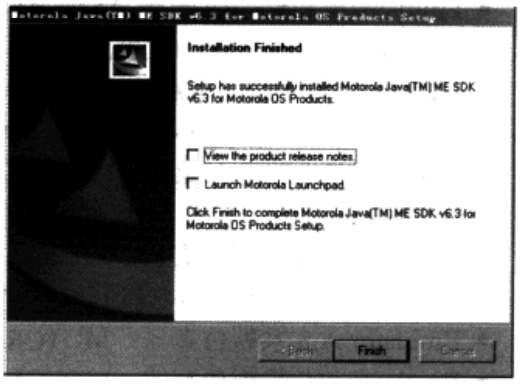


图 19-6 安装完成对话框

单击 Finish 按钮，完成 SDK 的安装。下面将介绍如何把这个 SDK 集成到 NetBeans 集成开发环境中。



19.2.2 在 NetBeans 中集成摩托罗拉 Java ME 开发包

本节虽然介绍的是如何集成摩托罗拉 Java ME 开发包，对于其他 Java ME 开发包的集成方法也适用。

在 NetBeans 集成开发环境中，选择“工具”→“Java 平台”命令，打开“Java 平台管理器”对话框，如图 19-7 所示。

单击“添加平台”按钮，打开“选择平台类型”对话框，如图 19-8 所示。这里选择“Java ME MIDP 平台仿真器”单选按钮。

Download at http://www.pin5i.com/

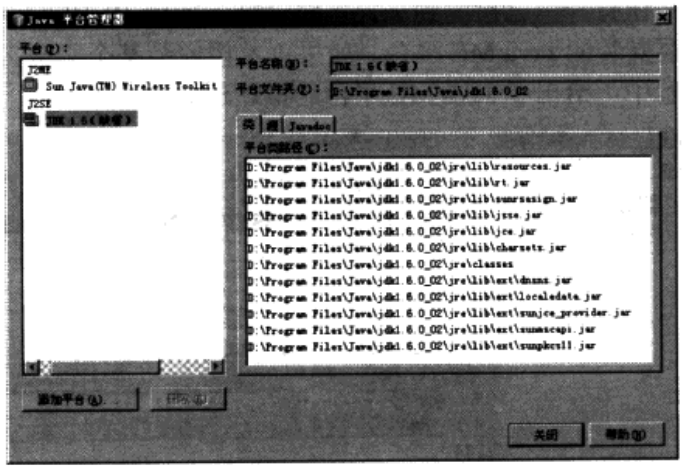


图 19-7 “Java 平台管理器”对话框

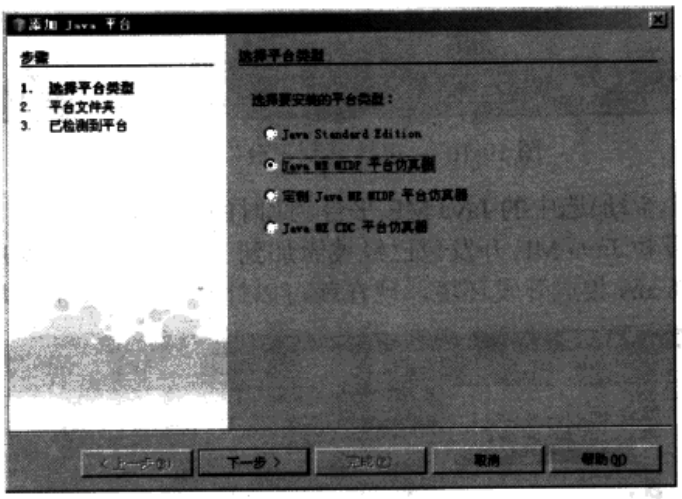


图 19-8 “选择平台类型”对话框

单击“下一步”按钮，打开“平台文件夹”对话框，如图 19-9 所示。在“选择要检测的平台”列表中自动列出了系统中安装的 Java ME 开发平台。选中旁边的复选框就检测开发平台是否可以被集成到 NetBeans 中。如果在“选择要检测的平台”列表中没有列出目标平台，可以单击“查找更多 Java ME 平台文件夹”按钮，手工选择 Java ME 平台。

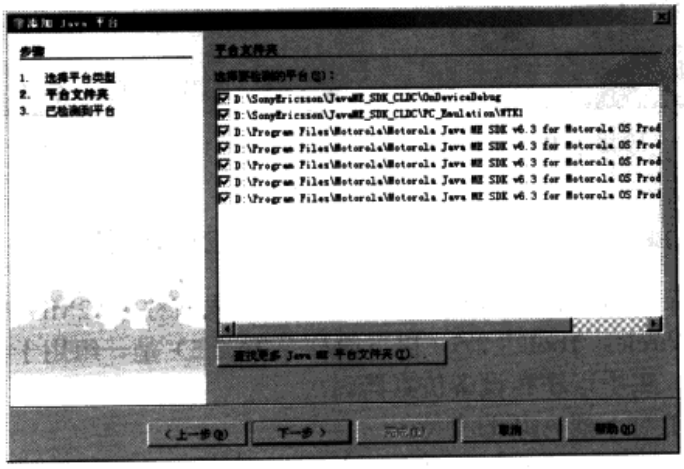


图 19-9 “平台文件夹”对话框

Download at http://www.pin5i.com/

单击“下一步”按钮打开“已检测到平台”对话框，如图 19-10 所示。在“选择要安装的平台”列表中列出了要安装的 Java ME 平台。如果不希望安装某个平台，则可以取消选中的复选框。在“描述”标签中，给出了 Java ME 平台的描述信息。

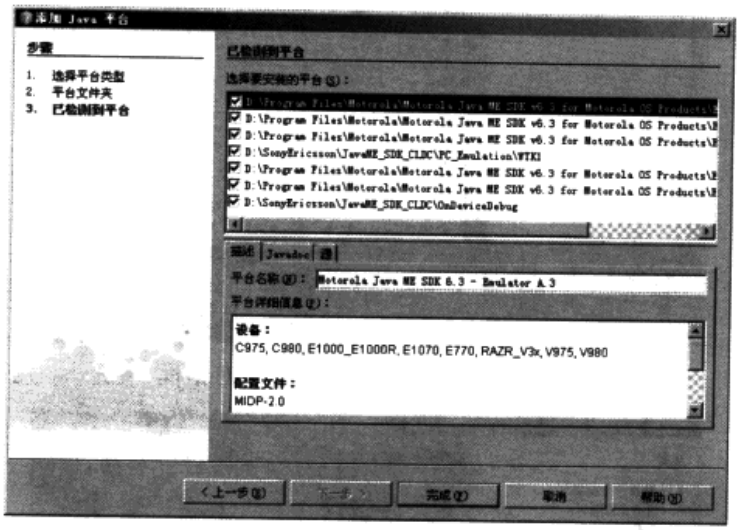


图 19-10 “已检测到平台”对话框

单击“完成”按钮，添加选中的 Java ME 平台。此时的“Java 平台管理器”对话框如图 19-11 所示。可以看到摩托罗拉 Java ME 开发包已经被添加到了 NetBeans 集成开发环境中。单击“关闭”按钮，返回 NetBeans 集成开发环境，现在就可以针对摩托罗拉的手机进行开发了。

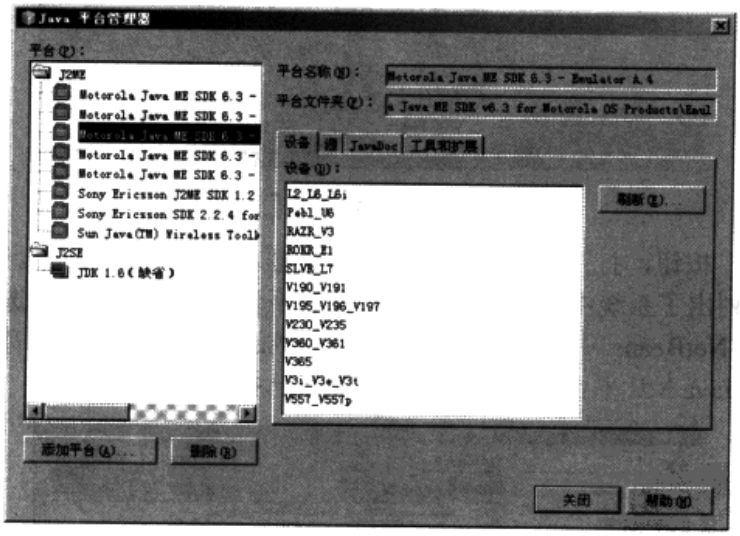


图 19-11 添加了摩托罗拉 Java ME 开发包的“Java 平台管理器”对话框



19.2.3 Sun Java Wireless Toolkit 2.5 for CLDC 简介

Mobility Pack 自带 Sun Java Wireless Toolkit 2.5.2（先前称为 J2ME Wireless Toolkit）手机仿真平台。Sun Java Wireless Toolkit 2.5（以下简称 WTK2.5）是一组用于创建 Java 应用程序的工具。包含生成工具、实用程序和设备仿真器。

WTK 2.5 版新增了对下列 API 的支持：

- Mobile Service Architecture（JSR 248）
- Security and Trust Services API for J2ME（JSR 177）

Download at http://www.pin5i.com/

- Location API for J2ME（JSR 179）
- SIP API for J2ME（JSR 180）
- Content Handler API（JSR 211）
- Scalable 2D Vector Graphics API for J2ME（JSR 226）
- Payment API（JSR 229）
- Advanced Multimedia Supplements（JSR 234）
- Mobile Internationalization API（JSR 238）
- Java Binding for the OpenGL(R) ES API（JSR 239）

在本书以后的示例中，将以 WTK 为仿真平台开发和调试 Java ME 应用程序。

19.3 创建一个简单的 Java ME 应用程序

本节将介绍如何开发一个简单的 Java ME 应用程序，并在摩托罗拉 Z3 手机仿真器上运行这个程序。

19.3.1 创建 Mobility 项目

在 NetBeans 集成开发环境中，选择“新建”→“新建项目”命令（Ctrl + Shift + N），打开“新建项目”对话框。在“类别”列表中选择 Mobility 节点，如图 19-12 所示。

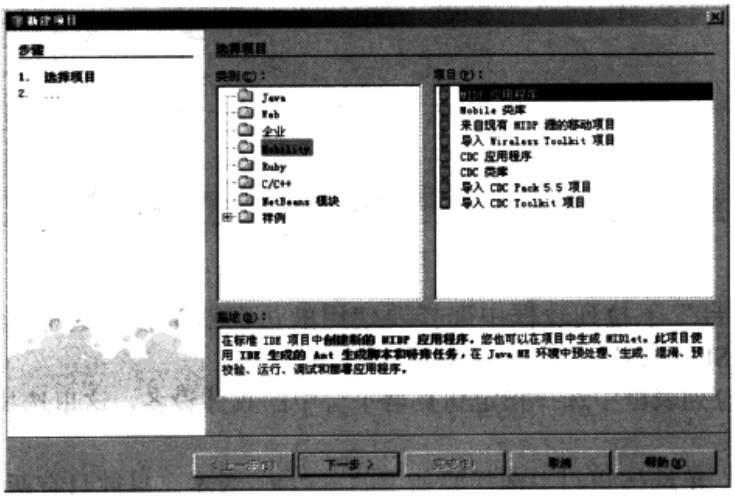


图 19-12 “新建项目”对话框的 Mobility 节点

在“项目”列表中选择“MIDP 应用程序”，单击“下一步”按钮打开“名称和位置”对话框，如图 19-13 所示。

在“项目名称”文本框中输入新项目的名称。在“项目位置”文本框中指定新项目文件夹的父文件夹，可以单击“浏览”按钮选择目标目录。集成开发环境将基于新项目名称来创建一个项目文件夹，该文件夹将存储于“项目位置”文本框中指定的文件夹中。在“项目文件夹”文本框中显示了最终创建的项目文件夹的路径。

如果希望新项目是主项目，则要选中“设置为主项目”复选框。如果选中“创建 Hello MIDlet”复选框，则可以自动在项目中创建一个“Hello MIDlet”Java ME 应用程序，该程序可以在打开的可视 Mobile 设计器中进行编辑。如果不选中“创建 Hello MIDlet”复选框，集成开发环境将创建空项目。这里为了演示创建 MIDlet 的具体过程，因此取消“创建 Hello MIDlet”复选框，

Download at http://www.pin5i.com/

下面将单独创建 Hello MIDlet 程序。单击“下一步”按钮，打开“选择默认平台”对话框，如图 19-14 所示。

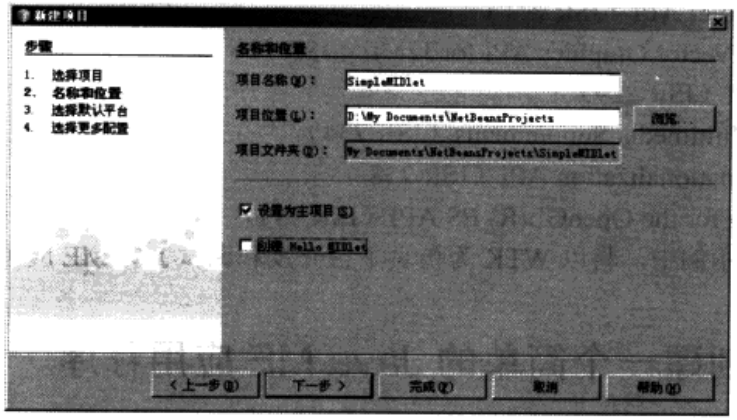


图 19-13 “名称和位置”对话框

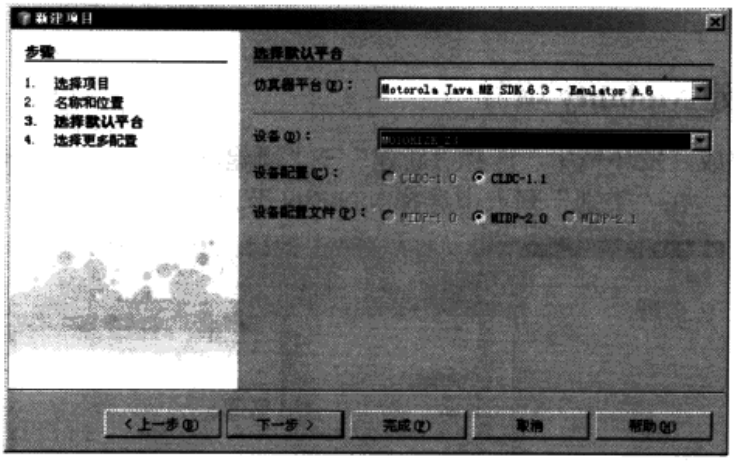


图 19-14 “选择默认平台”对话框

在“仿真器平台”下拉列表框中选择要使用的仿真器平台，这里选择上一节中安装的“Motorola Java ME SDK 6.3”开发包的 A.6 仿真器平台。

在“设备”下拉列表框中选择指定仿真器平台中的具体设备，这里选择 A.6 仿真器平台中的 Motorola Z3 手机。

在“设备配置”单选按钮组中选择要使用的 CLDC 版本。目前只有两个版本：CLDC 1.0 和 CLDC 1.1。一般比较新的手机都支持 CLDC 1.1 版本。大多数的平台都有默认的 CLDC 版本，一般情况下选择默认的设置即可。



技巧：在实际开发中，开发人员应该清楚 CLDC 1.0 版本和 CLDC 1.1 版本有哪些具体的不同。CLDC 1.1 版本向下兼容 CLDC 1.0 版本。在自己的项目中，如果只用到了 CLDC 1.0 的 API，那么可以在这里选择 CLDC 1.0 版本（有些仿真器平台是可以选择的），而不必过分追求高版本。这样开发出来的应用程序可能会在 CLDC 1.0 版本的手机上运行，从而支持更多的手机机型。

在“设备配置文件”单选按钮组中选择要使用的 MIDP 版本。目前只有两个版本：MIDP 1.0 和 MIDP 2.0。MIDP（Mobile Information Device Profile）是 CLDC 的一个补充，为嵌入式设备提供了包括用户界面、计时器、永久存储记录、HTTP 网络连接等功能。一般比较新的手机都

Download at http://www.pin5i.com/

支持 MIDP 2.0 版本。大多数的平台都有默认的 MIDP 版本，一般情况下选择默认的设置即可。

单击“下一步”按钮，打开“选择更多配置”对话框，如图 19-15 所示。单击“完成”按钮创建一个空白的 Mobility 项目，即一个 MIDlet 套件。在手机上可以运行 Java ME 程序又被称为 MIDlet，一个 MIDlet 套件（MIDlet Suit）中可以包含多个 MIDlet。

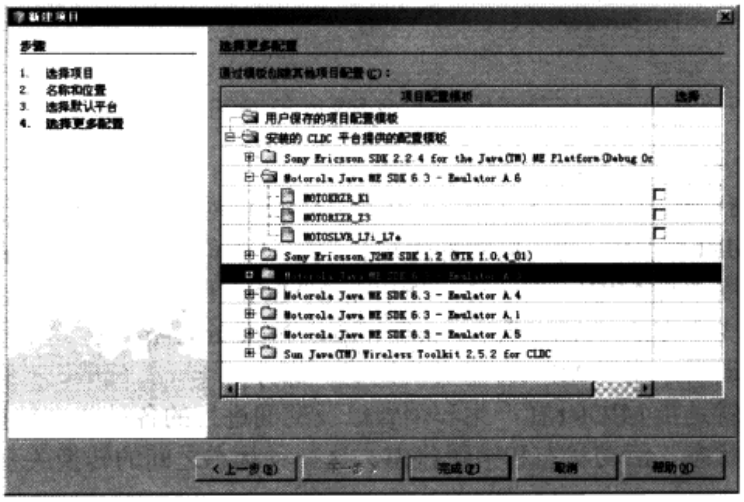


图 19-15 “选择更多配置”对话框



19.3.2 创建 MIDlet

在“项目”窗口中，右键单击 SimpleMIDlet 项目节点，选择弹出菜单中的“新建”→“MIDlet”命令，打开“新建文件”对话框，如图 19-16 所示。

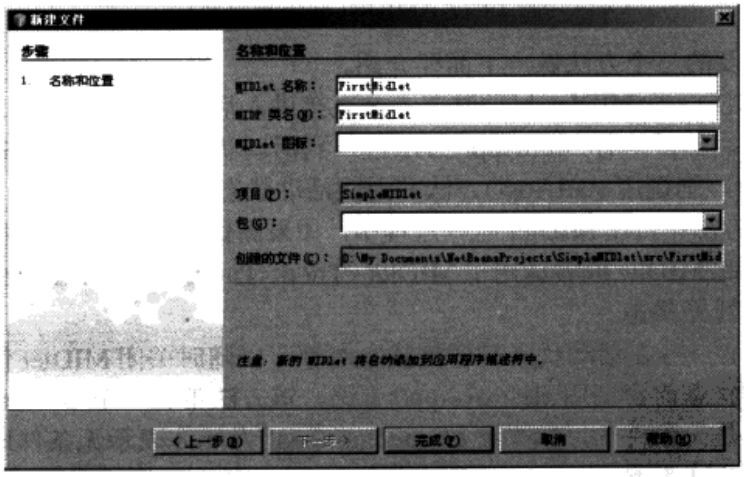


图 19-16 “新建文件”对话框

在“MIDlet 名称”文本框中输入新建的 MIDlet 的名称“FirstMidlet”。在“MIDP 类名”文本框中输入此 MIDlet 的类名，在输入 MIDlet 名称的时候，会默认把 MIDlet 的类名设置为 MIDlet 的名字。一般情况下不需要再修改。

如果应用程序中添加了图像，可以在“MIDlet”图标下拉列表框中选择。在“包”文本框中输入此 MIDlet 类所属的包，单击“完成”按钮。

NetBeans 为新建的 MIDlet 自动生成了一些代码，下面是 FirstMIDlet.java 的完整源代码(注释已略去)：

Download at http://www.pin5i.com/


```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class FirstMidlet extends MIDlet {
    public void startApp() {
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
}
```

在上面这段代码中首先导入用于开发 MIDP 应用程序的包，MIDlet 类位于 java.microediton.midlet.*包中，而一些图形用户界面组件类位于 javax.microedition.lcdui.*包中。每一个 Java ME 程序都是 MIDlet 的子类，所以 FirstMidlet 也继承了 MIDlet 类。

在类中生成了 3 个方法：startApp()、pauseApp()和 destropApp()，这 3 个方法在 MIDlet 的状态发生变化时被调用，属于 MIDlet 生命周期的一部分。

所谓生命周期就是指 MIDlet 在产生和销毁以及期间进行的各种状态转换的全过程。MIDlet 有三种状态：暂停状态、活动状态和销毁状态，这三种状态之间的转换关系如图 19-17 所示。

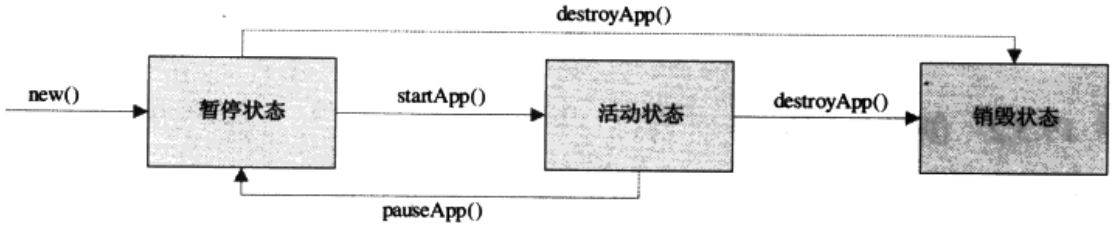


图 19-17 MIDlet 的生命周期

下面对图中涉及的三个方法做一下简要介绍。

- startApp()方法：当 MIDlet 启动时并不是直接进入活动状态而是进入暂停状态，这个过程非常短暂，然后执行 startApp()方法进入到活动状态。在暂停状态时，如果用户手工恢复 MIDlet，则调用 startApp()方法并返回活动状态。
- pauseApp()方法：当在 MIDlet 运行过程中，如果高优先级的外部事件到来（例如来电）、用户手工挂起 MIDlet 或者调用了 MIDlet 的 notifyPaused 方法，MIDlet 将调用 pauseApp()方法并进入暂停状态。
- destroyApp()方法：在活动状态和暂停状态时都可以随时关闭 MIDlet 使其进入死亡状态。在进入死亡状态前会调用 destroyApp()方法，该方法带有一个 boolean 类型的参数，指示销毁是否是无条件的，如果参数 unconditional 为 true 表示无条件地将 MIDlet 销毁；当 unconditional 为 false 时，MIDlet 可以通过抛出 MIDletStateChangeException 异常来保持它当前的状态。

下面创建一个简单的 Java ME 程序，向 FirstMidlet 中加入一些代码，最终代码如下所示：

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class FirstMidlet extends MIDlet {
    private Display display;//声明显示对象 display
    private Form form;//声明表单 form

    public FirstMidlet(){
        display = Display.getDisplay(this);//获得当前 MIDlet 的显示对象
```

Download at http://www.pin5i.com/


```
        form = new Form("Welcome");//创建 form 对象并设置其标题为“Welcome”
        form.append("Enjoy yourself!");//向 form 中加入一些文字
    }
    public void startApp() {
        display.setCurrent(form);//设置在启动 MIDlet 时显示 form
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
}
```

这里解释一下 Display，每个 MIDlet 都有且仅有一个对应的 Display 对象，Display 代表系统的显示和输入设备的管理器，利用 Display 对象可以设置当前屏幕显示的各种 Displayable 对象。运行刚刚创建的 SimpleMIDlet，单击工具栏上的“运行主项目”按钮，或者按下快捷键 F6，NetBeans 会启动 Motorola A.6 仿真器运行 MIDlet，运行结果如图 19-18 所示。



图 19-18 程序运行结果

19.4 Java ME 应用程序结构分析

Java ME 应用程序的结构和 Java 应用程序的结构略有不同，可以在“文件”窗口中查看其结构。在生成完项目之后 NetBeans 创建了如图 19-19 所示目录和文件。

- build.xml 文件：编译、运行以及调试项目所需的 Ant 脚本。
- nbproject 目录：存放项目相关数据文件，用于产生项目的生成脚本。
- src 目录：存放 Java 源文件。
- dist 目录：存放项目打包文件（JAR 文件）和应用程序描述符文件（JAD 文件）。
- build 目录：包含如下目录。
 - complied 目录：包含所有编译的类。

Download at http://www.pin5i.com/

- preprocessed 目录：存放预处理后的源文件。如果您使用的是项目配置（关于项目配置的内容会在本章后面的部分介绍），则这些文件与原始源文件会有所不同。
- obfuscated 目录：存放类文件的混淆版本。
- preverified 目录：存放类文件的预校验版本。这些文件将打包到项目的分发 JAR 文件中。

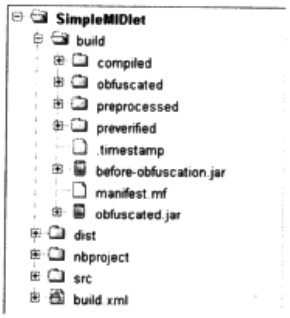


图 19-19 Java ME 应用程序结构

19.5 配置 Java ME 应用程序

从 Java ME 应用程序的创建到生成的各个步骤，都可能需要对其进行不同配置。要配置 Java ME 应用程序可以在“项目”窗口中右键单击要配置的项目节点，选择弹出菜单中的“属性”命令，打开“项目属性”对话框，下面的所有设置都是在这个对话框中进行的。

19.5.1 查看项目常规属性

在“项目属性”对话框中可以查看和项目有关的属性信息。选择“常规”节点，如图 19-20 所示。“项目名称”文本框中显示了项目的名称；“项目文件夹”文本框中显示了项目所在文件夹；“项目源位置”文本框中显示了源文件所存放的位置；“所需的项目”列表中显示了当前项目所依赖的所有项目的列表。这些信息都是不可以更改的。如果选择“重新生成所需的项目”复选框，则会在生成项目之前重新生成所需的项目，否则将不会重新生成所需的项目。

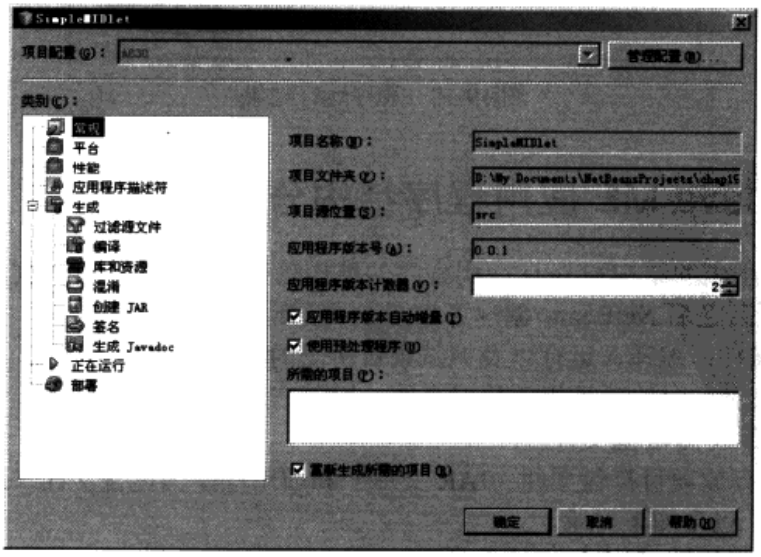


图 19-20 查看项目常规属性

Download at http://www.pin5i.com/

19.5.2 定制平台

在开发过程中为了保证应用程序的良好移植性，往往需要在不同的仿真平台上进行测试，可以在“项目属性”对话框中选择不同的仿真平台并选择需要的可选 API。如图 19-21 所示。

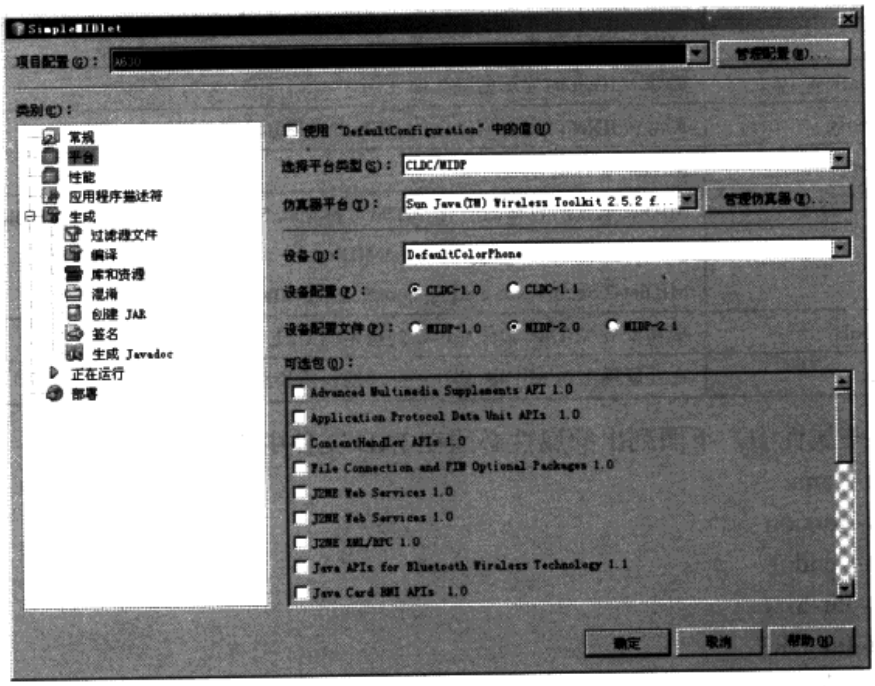


图 19-21 定制平台

在“仿真器平台”下拉列表框中选择已在集成开发环境中注册的仿真器平台，也可以单击“管理仿真器”按钮按照本章前面介绍的方法添加其他平台。

可以在“设备”下拉列表框中选择平台所带的仿真设备，同时可以通过单击单选按钮选择“设备配置”的版本以及“设置配置文件”的版本。在“可选包”列表中列出了设备支持的可选包，可以根据需要进行选择。

19.5.3 设置应用程序描述符属性

当 Java ME 应用程序发布时会被打包成 JAR 文件，这个文件被称为 MIDlet 套件。在这个 JAR 文件中有个 Manifest.mf，它和一个可选的应用程序描述符文件即 JAD 文件一起用来描述套件属性信息。MIDP 专家组所定义的属性是以“MIDlet-”为前缀的，开发人员也可以自定义属性。这些属性有些必须在 JAD 文件和 Manifest.mf 文件中定义，有些为可选的。表 19-1 列出了常用预定义的 MIDlet 属性。

表 19-1 MIDlet 属性

属性名称	描 述
MIDlet-Name	MIDlet 程序的名称
MIDlet-Version	MIDlet 套件版本号
MIDlet-Vendor	MIDlet 套件提供商名称
MIDlet-Jar-Size	JAR 文件的大小，以 byte 为单位
MIDlet-Jar-URL	下载 JAR 文件的地址

Download at http://www.pin5i.com/

续表

属性名称	描　述
MicroEdition-Configuration	要求的配置版本，目前可以为 CLDC-1.0 或者 CLDC-1.1
MicroEdition-Profile	要求的简表版本，目前可以填写为 MIDP-2.1、MIDP-2.0 或者 MIDP-1.0
MIDlet-Icon	MIDlet 程序的图标，要求为 PNG 文件
MIDlet-Data-Size	MIDlet 持续存储的数据区域大小
MIDlet-Delete-Confirm	删除 MIDlet 时显示的确认信息
MIDlet-Delete-Notify	删除 MIDlet 时可以向指定的 URL 发送通知信息
MIDlet-Description	MIDlet 套件的详细说明
MIDlet-<n>	MIDlet 套件所包含的所有 MIDlet，标记其 MIDlet 名称、图标和 MIDlet 的类名。例如， MIDlet-1:First,first.png,com.FirstMIDlet MIDlet-2:Second,second.png,com.SecondMIDlet
MIDlet-Install-Notify	成功安装 MIDlet 套件后可以向指定的 URL 发送通知信息
MIDlet-Info-URL	在互联网上对 MIDlet 套件进行详细说明的地址

在以上这些属性中，下面列出的属性必须在 JAD 文件中定义：

- MIDlet-Name
- MIDlet-Version
- MIDlet-Vendor
- MIDlet-Jar-Size
- MIDlet-Jar-URL

下面列出的属性必须在 Manifest.mf 文件中定义：

- MIDlet-Name
- MIDlet-Version
- MIDlet-Vendor

下面列出的属性必须在 JAD 文件或 Manifest.mf 文件中定义：

- MIDlet-<n>
- MicroEdition-Configuration
- MicroEdition-Profile

在 NetBeans 集成开发环境中可以方便地设置以上这些属性，在“项目属性”对话框中，选择“应用程序描述符”节点下的“属性”标签，如图 19-22 所示。

在“JAD 和 Jar 清单的常规属性”列表中列出一些属性。要添加新的属性，单击“添加”按钮，弹出“添加属性”对话框，如图 19-23 所示。

在“名称”下拉列表框中选择预定义的属性名称，也可以输入自定义的属性名称，在“值”文本框中输入属性的值。如果是自定义的属性，还可以在“位置”下拉列表框中选择该属性被添加到的位置，可以是“JAD 和清单”、“JAD”和“清单”，“JAD”指的是生成的 JAD 文件；“清单”指的是 Manifest.mf 文件。填写完成后，单击“确定”按钮返回。

可以选择列表中的一个属性，单击右侧的“编辑”按钮可以在打开的“编辑属性”对话框中编辑该属性，如果在编辑“MIDlet-Version 属性”时，在“值”文本框中输入“${deployment number}”，则会在每次部署应用程序时自动在 JAR 文件中递增 MIDlet-Version 属性的值。这对移动设备检查是否存在已安装的应用程序更新非常有用。通过递增 MIDlet-Version 属性的值，可以让此类设备在你部署应用程序时自动检测新版本。要删除某个属性，在列表中选中它，然后单击“删除”按钮即可，注意必需的属性是不能被删除的。

Download at http://www.pin5i.com/

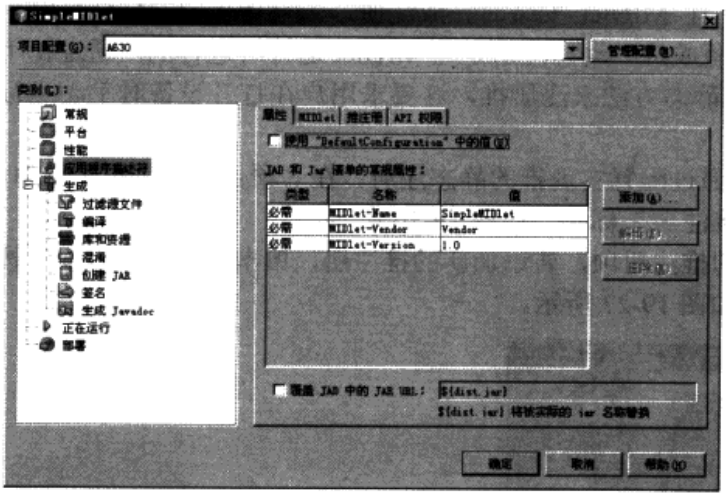


图 19-22 设置应用程序描述符属性

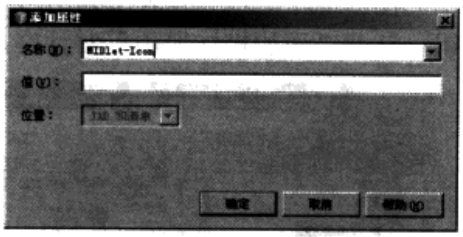


图 19-23 “添加属性”对话框

19.5.4 设置 MIDlet

在上面的属性设置中无法设置 MIDlet-<n>的属性，该属性要在“应用程序描述符”节点下的“MIDlet”标签下进行设置，如图 19-24 所示。

在“套件中的 MIDlet”列表中列出当前 MIDlet 套件中的所有 MIDlet，单击右侧的“添加”按钮打开“添加 MIDlet”对话框，如图 19-25 所示。在“MIDlet 名称”文本框中输入要添加的 MIDlet 的名称，在“MIDlet 类”下拉列表框中选择 MIDlet 类，如果项目绑定了图像文件（关于如何将图像文件添加到当前项目中将在本章后面部分讲解），则可以在“MIDlet 图标”下拉列表框中为当前 MIDlet 选择一个图像文件作为图标。分别进行“添加”、“编辑”和“删除”操作。单击“上移”和“下移”按钮可以调整 MIDlet 顺序，这个顺序和 MIDlet-<n>中的 n 的序号相对应。

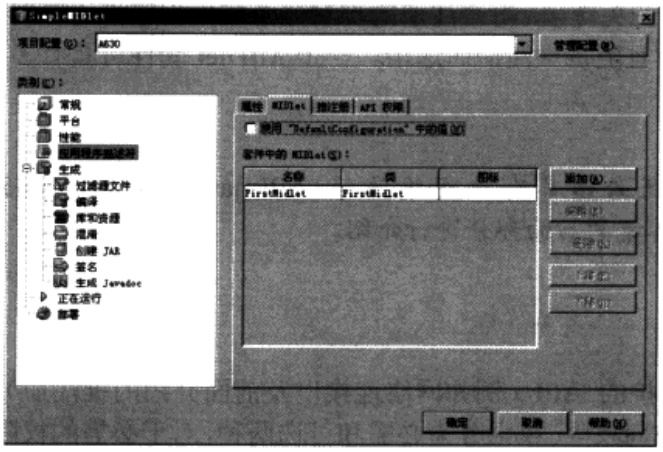


图 19-24 设置 MIDlet

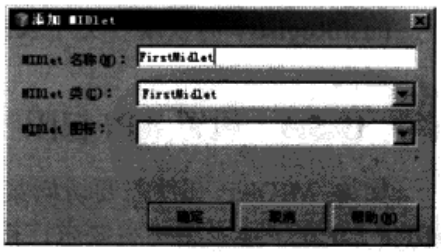


图 19-25 “添加 MIDlet”对话框

19.5.5 设置推注册

推技术是一种通过异步方式将信息传送至设备并自动启动 MIDlet 的机制。这个技术是 MIDP 2.0 的一个可选项，使用时一定要查看设备是否支持。要使用推技术，则必须将其进行注册，注册分静态注册和动态注册两种方式。

Download at http://www.pin5i.com/

静态注册一般是基于网络连接的，在 MIDlet 安装时完成注册，当有传入的网络连接时系统将启动已经进行推注册的 MIDlet 进行响应。动态注册是在 MIDlet 运行时进行的，通过使用 javax.microedition.io.PushRegistry 中的静态方法来注册推，这要求用户在打开设备时手动运行 MIDlet。

在 NetBeans 集成开发环境中可以通过配置实现推的静态注册。在“项目属性”对话框中，选择“应用程序描述符”节点下的“推注册”标签，如图 19-26 所示。

在“推注册项”列表中列出所有的推注册项，要添加新的推注册，单击右侧的“添加”按钮，打开“添加推注册项”对话框，如图 19-27 所示。

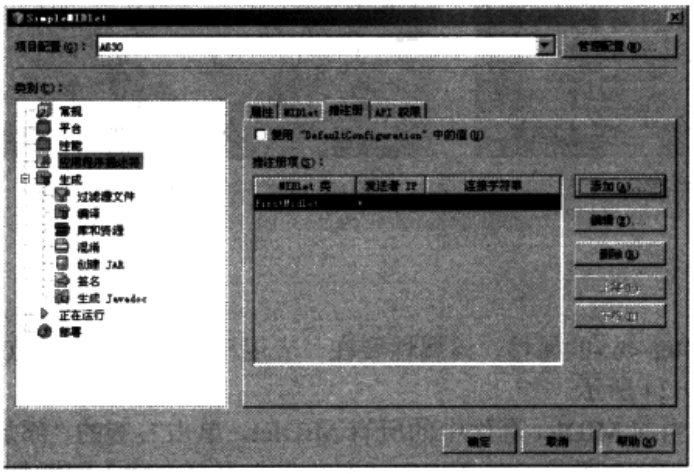


图 19-26 设置推注册

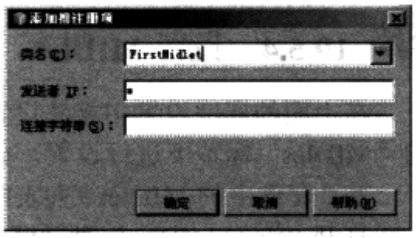


图 19-27 添加推注册项

在“类名”下拉列表框中选择 MIDlet 类。在“发送者 IP”文本框中输入要响应的 IP，如果该值为通配符*，则接受来自任何 IP 的连接。如果使用数据报或套接字连接，则其值可以为数字 IP 地址。在“连接字符串”文本框中输入标识连接协议与端口号的连接字符串。单击“确定”按钮返回“项目属性”对话框。

NetBeans 会为每个注册的推指派唯一的关键字，每个注册的推对一个应用程序描述符属性 MIDlet-Push-<n>，其中序号从 1 开始计数且随每次注册而递增。一个 MIDlet 套件可具有多个推注册。



说明：要使用推注册，您还必须具有访问推注册 API javax.microedition.io.PushRegistry 的权限。关于权限将在本章的后面部分进行介绍。



19.5.6 设置 API 权限

如果 MIDlet 套件需要访问某些受保护的 API（例如网络连接以及前面介绍的推注册），MIDP 2.0 安全体系结构将要求连接授予访问权限。权限分为必需和可选两种，对于必需的权限，在安装时，如果应用程序管理软件不能授予所需的权限，则不会安装应用程序。对于可选的权限，如果权限被拒绝，应用程序可以继续执行，但其功能可能会受到限制。必需的权限和可选的权限对应的应用程序描述符属性分别为 MIDlet-Permissions 和 MIDlet-Permissions-Opt。

在 NetBeans 集成开发环境中，要设置 API 权限，在“项目属性”对话框中选择“应用程序描述符属”节点下的“API 权限”标签，如图 19-28 所示。

要添加 API 权限，单击右侧的“添加”按钮，打开的“添加 API 权限”对话框，如图 19-29 所示。从“API 包”下拉列表框中选择需要设置权限的 API 包，单击“完成”按钮返回。在“请

Download at http://www.pin5i.com/

求的权限”列表中列出了所有请求的权限，如果选中“必需”复选框，则此 API 权限为必需的权限，否则为可选的权限。可以单击“删除”按钮删除不需要的 API 权限。

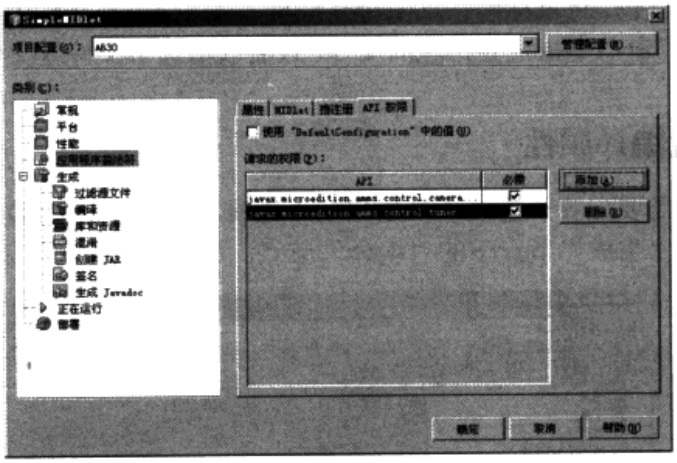


图 19-28　设置 API 权限

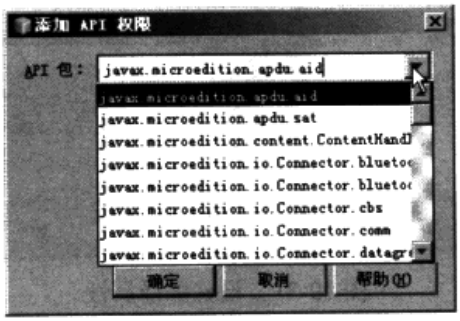


图 19-29 “添加 API 权限”对话框

19.5.7　过滤源文件

有时在开发过程中需要的文件而在生成时不需要它，这时可以将其过滤掉。在“项目属性”对话框中选择“生成”节点下的“过滤源文件”节点，如图 19-30 所示。

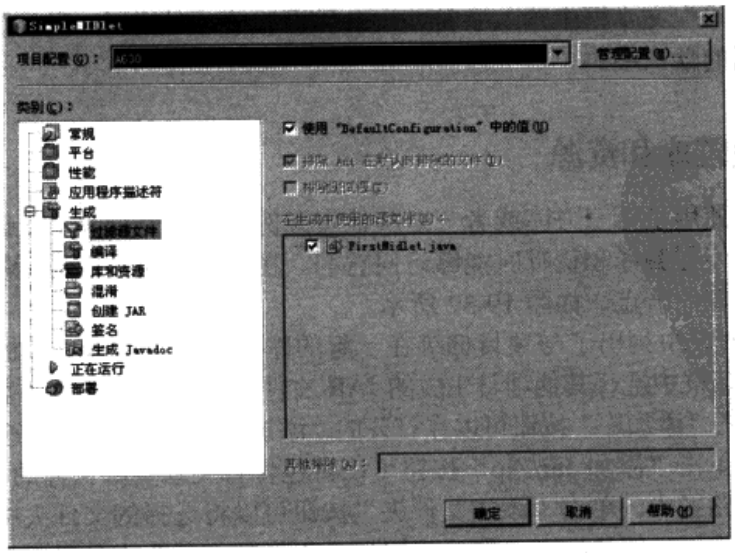


图 19-30　过滤源文件

Download at http://www.pin5i.com/

如果选中“排除 Ant 在默认时排除的文件”复选框，则从 JAR 文件中排除由 Ant 定义且在默认时排除的文件（如版本控制文件和文件夹）。如果选中“排除测试源”复选框，则从 JAR 文件中排除测试和测试包。将鼠标停在这两个复选框上方，会出现排除的文件列表。可以在“在生成中使用的源文件”窗口中指定要排除的文件，只需取消选中文件或文件夹前的复选框即可。



19.5.8 设置编译属性

在编译过程有时需要配置一些编译参数和编译选项，这些参数和选项可以在“项目属性”对话框中设置。选择“生成”节点下的“编译”节点，如图 19-31 所示。

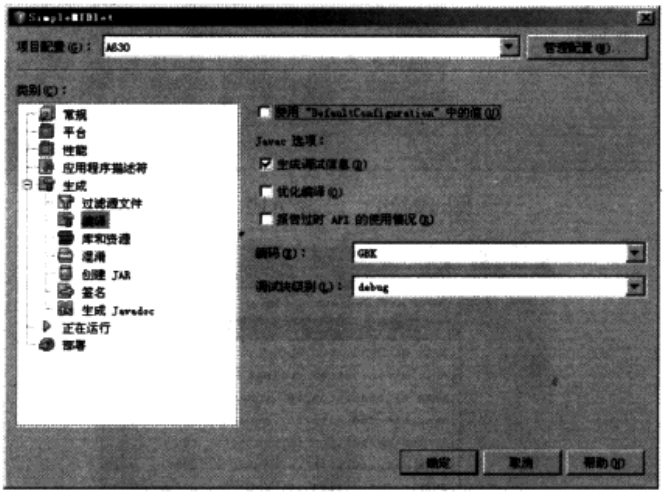


图 19-31 设置编译属性

如果选择“生成调试信息”复选框，编译器将生成行号和源文件信息，这其实就是 javac 命令中的-g 选项。如果未选择，则不生成调试信息（-g:none）。如果选择“优化编译”复选框，则会优化编译的应用程序以便于执行，这其实就是 javac 命令中的-O 选项。优化操作可能会减慢编译速度，生成更大的类文件，并使程序难以调试。如果选择“报告过时 API 的使用情况”复选框，则编译器会列出过时的成员或类的每次使用或覆盖的信息，这其实就是 javac 命令中的-deprecated 选项。如果未选中，则编译器只显示使用或覆盖过时成员或类的源文件名。

还可以在“编码”文本框中输入字符编码名称设置编译时编码，默认为 JVM 和 NetBeans 集成开发环境使用的默认编码。



19.5.9 设置库和资源

如果在项目中使用了第三方库或者一些其他资源例如图像文件等，则必须将它们添加到项目的类路径中，以便在编译和执行时能够访问它们。在“项目属性”对话框中，选择“生成”节点下的“库和资源”节点，如图 19-32 所示。

“库和资源”列表中列出了与项目捆绑在一起的库和资源。单击“添加项目”按钮在打开的“选择项目”对话框中选择其他项目生成的 JAR 文件，这样此 JAR 文件就被添加到当前项目的类路径中了。单击“添加库”按钮可以在打开的“添加库”对话框中将已安装的库添加到当前项目的类路径中。单击“添加 Jar/Zip”按钮可以将包含有类或资源文件的 JAR 文件或 ZIP 文件添加到项目的类路径中。单击“添加文件夹”按钮可以将选择的文件夹中的所有文件添加到当前项目的类路径中。一般情况下，在项目根目录下新建“res”文件夹，将项目所需的资源文件如图像和声音文件等存放在里面，然后再单击“添加文件夹”按钮将其添加到项目类路径中。

Download at http://www.pin5i.com/

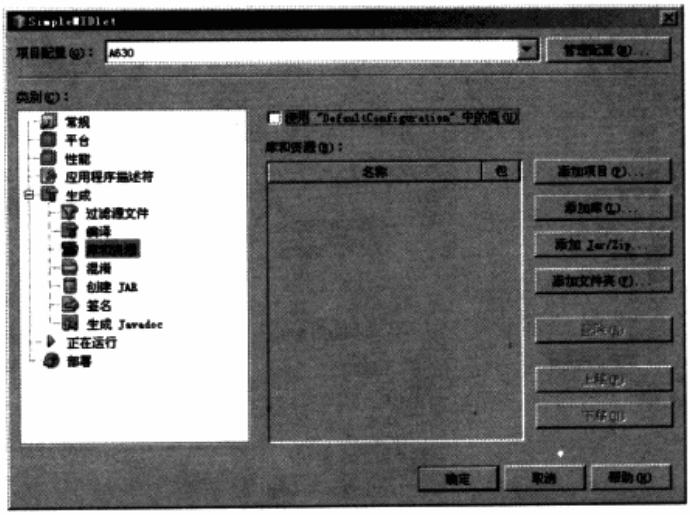


图 19-32 设置库和资源

要更改库或资源在类路径中的顺序，请选择列表中相应的项，然后单击“上移”和“下移”按钮。要删除列表中的项，先将其选中然后单击“删除”按钮即可。



19.5.10 设置 JAR 和 JAD 文件

在“项目属性”对话框中可以设置在生成项目时创建的 JAR 文件和 JAD 文件的文件名。选择“生成”下的“创建 JAR”节点，如图 19-33 所示。

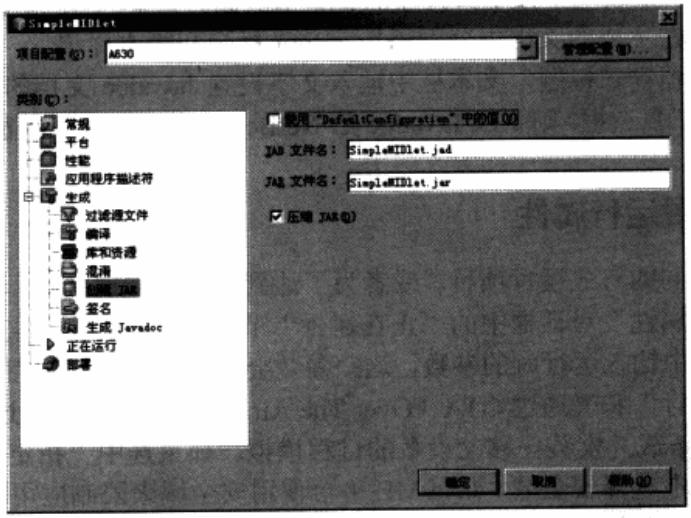


图 19-33 设置 JAR 和 JAD 文件

在“JAD 文件名”文本框中输入生成项目时的 JAD 文件名，在“JAR 文件名”文本框中输入生成项目时的 JAR 文件名，如果选择“压缩 JAR”复选框可以压缩 JAR 文件。



19.5.11 设置 Javadoc 生成

可以在“项目属性”对话框中配置 NetBeans 集成开发环境生成 Javadoc 文档的方式。选择“生成”节点下的“生成 Javadoc”节点，如图 19-34 所示。

Download at http://www.pin5i.com/

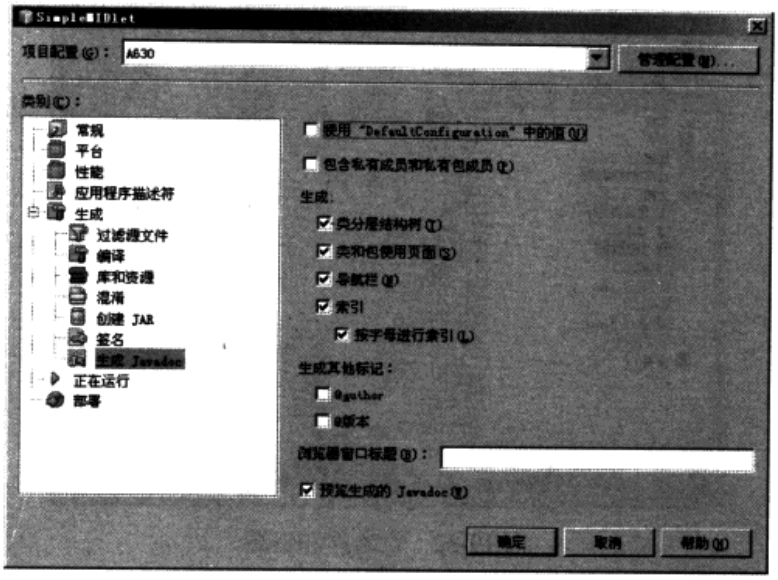


图 19-34 设置 Javadoc 生成

如果选择“包含私有成员和私有包成员”复选框，将在生成的 Javadoc 文档中包含在源文件中声明为私有的成员和私有包成员。

如果选择“类分层结构树”复选框，则可以生成显示类分层结构树的页面。如果选择“类和包使用页面”复选框，则可以生成显示类和包使用情况的页面。如果选择“导航栏”复选框，则可以在生成的页面的顶部和底部显示导航栏。如果选择“索引”复选框并选择“按字母进行索引”复选框，则可以生成一个按字母排序的类索引页面。如果选择“@author”复选框和“@版本”复选框，则将在生成的文档中分别添加 author 标记和版本标记。

还可以在“浏览器窗口标题”文本框中输入文字设置 Javadoc 文档的网页标题。如果选择“预览生成的 Javadoc”复选框，则生成 Javadoc 后 NetBeans 会自动打开浏览器显示生成的文档。



19.5.12 设置运行属性

有时需要使用不同的方式运行项目，或者为了达到不同的测试效果，需要设置运行时的参数。这可以在“项目属性”对话框中的“正在运行”节点设置，如图 19-35 所示。在“仿真器命令行选项”文本框中输入运行时的参数，这些参数会随仿真器的不同而不同。项目的执行方式可以选择“常规执行”和“通过 OTA（Over The Air 部署）执行”，通过 OTA 执行，实际上就是对 MIDlet 套件下载并安装在移动设备的过程模拟。如果选中“指定安全域”复选框，可以在其后的下列列表中选择安全域，Java ME 平台使用安全域来限制应用程序对受保护 API 的访问，但是安全域因设备的不同而有所不同。



19.5.13 设置部署方法

部署是为了进一步进行测试或实现而将集成开发环境中的 MIDP 应用程序传输到移动设备或服务器的过程。可以在“项目属性”对话框中设置应用程序部署的方法，选择“部署”节点，可以在“选择部署方法”下拉列表框中选择需要的部署方法，下面对几种部署方法分别加以介绍。

Download at http://www.pin5i.com/

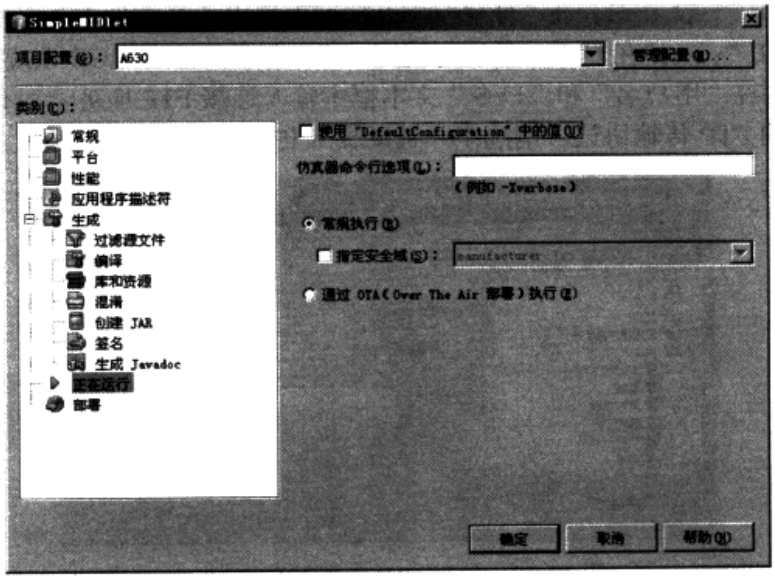


图 19-35 设置运行属性

1. 文件复制方法

可以将 MIDlet 套件中的 JAR 和 JAD 文件移到本地系统中的指定位置。在“选择部署方法”下拉列表框中选择“文件复制”，如图 19-36 所示。在“目标目录”中输入要将 MIDlet 套件复制到的目标目录的位置，也可以单击右侧的“浏览”按钮进行选择。

2. 使用 FTP 方法

可以使用文件传输协议（File Transfer Protocol, FTP）协议方法，将 MIDlet 套件中的 JAR 与 JAD 文件通过互联网传输到远程服务器上的指定位置。在“选择部署方法”下拉列表框中选择“文件传输协议（FTP）”，如图 19-37 所示。

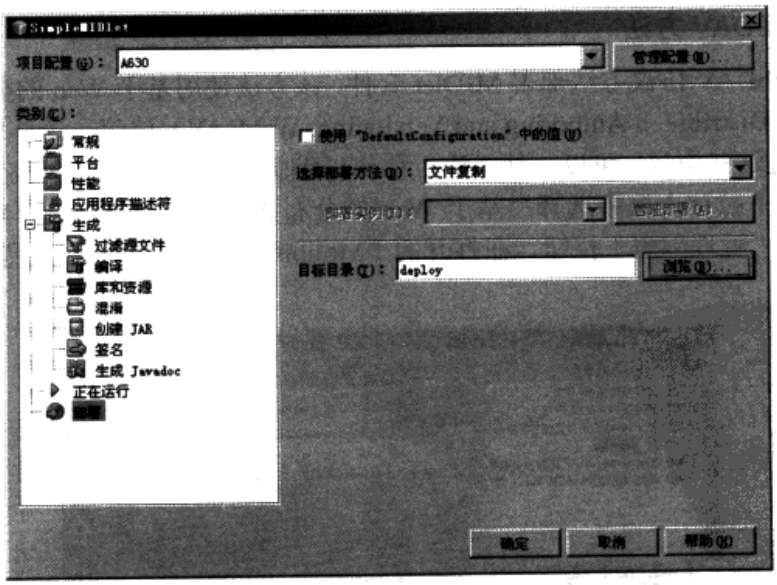


图 19-36 使用文件复制部署 MIDlet 套件

在“部署实例”下拉列表框中选择部署实例。在“目标目录”文本框中输入要部署到远程服务器的目录。可以单击“部署实例”后的“管理部署”按钮，在打开“Mobility 部署”对话框中设置部署实例，如图 19-38 所示。在服务器文本框中输入服务器名称或 IP 地址，在“端口”

Download at http://www.pin5i.com/

文本框中输入端口，一般为 21。在“分隔符”下拉列表框中选择文件位置名称分隔符是使用斜杠“/”还是反斜杠“\”（如在 UNIX 系统中使用“/”，在 Windows/DOS 系统中使用“\”）来分隔目录名称。分别在“用户名”和“口令”文本框中输入连接 FTP 服务器时使用的用户名和口令。要使用被动的 FTP 传输协议，请选中“被动模式传输”复选框。

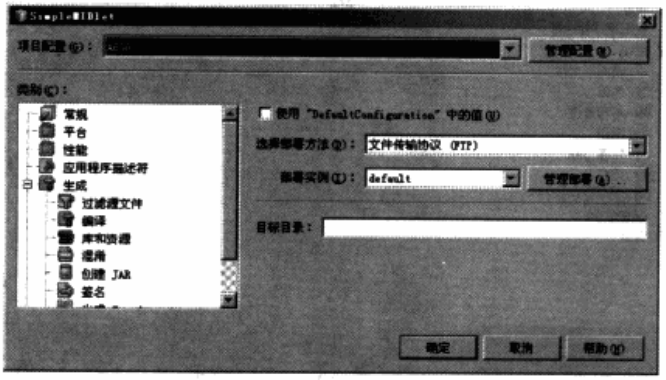


图 19-37 使用 FTP 部署 MIDlet 套件

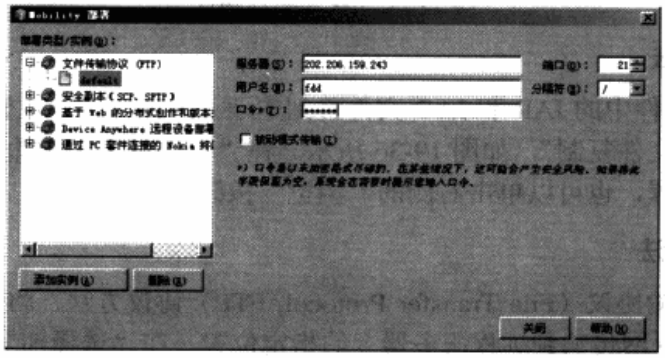


图 19-38 “Mobility 部署”对话框

3. 使用 WebDAV 方法

可以使用 WebDAV 协议方法部署 MIDlet 套件，该方法使用基于 Web 的分布式编制和版本控制（Web-based Distributed Authoring and Versioning, WebDAV）协议，这些协议是一组 HTTP 协议的扩展。通过这些协议，可以与他人进行协作以编辑和管理远程 Web 服务器上的文件。在“选择部署方法”下拉列表框中选择“基于 Web 的分布式创作和版本控制（WebDAV）”。单击“部署实例”后的“管理部署”按钮，在打开的“Molibity 部署”对话框中设置部署参数，如图 19-39 所示。

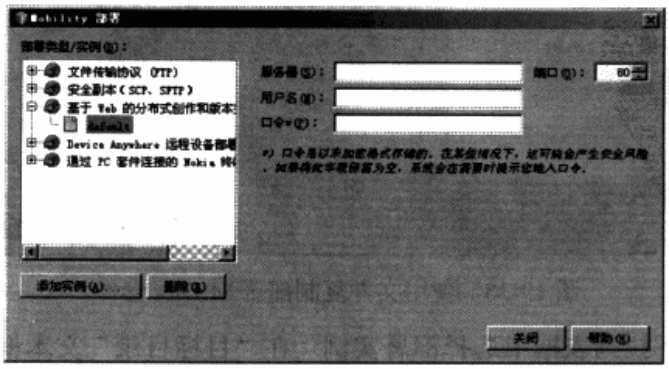


图 19-39 使用 WebDAV 部署 MIDlet 套件

Download at http://www.pin5i.com/

在“服务器”文本框中输入服务器名称或 IP 地址，然后分别在“用户名”和“口令”文本框中输入用户名以及用户口令。

4. 使用 SCP 或 SFTP 方法

可以使用安全客户端协议（Secure Client Protocol, SCP）或安全文件传输协议（Secure File Transfer Protocol, SFTP）部署方法建立安全连接并对数据进行加密，然后在客户端与服务器之间传输数据。在“选择部署方法”下拉列表框中选择“安全副本（SCP、SFTP）”，在“服务器”文本框中输入目标服务器的名称或 IP 地址。单击“部署实例”后的“管理部署”按钮，在打开的“Molibity 部署”对话框中设置部署参数，如图 19-40 所示。在“目标目录”文本框中输入要将 MIDlet 套件发送到的服务器位置。在“端口”文本框中输入服务器的 SCP 连接端口。

在“用户名”文本框中输入 SCP 登录的名称。如果使用口令验证，则在“口令”文本框中输入部署 MIDlet 时所需的口令；如果使用密钥身份验证，则要在“密钥文件”文本框中输入一个密钥文件的路径并且在“密钥口令短语”文本框中输入部署 MIDlet 时所需的口令短语。

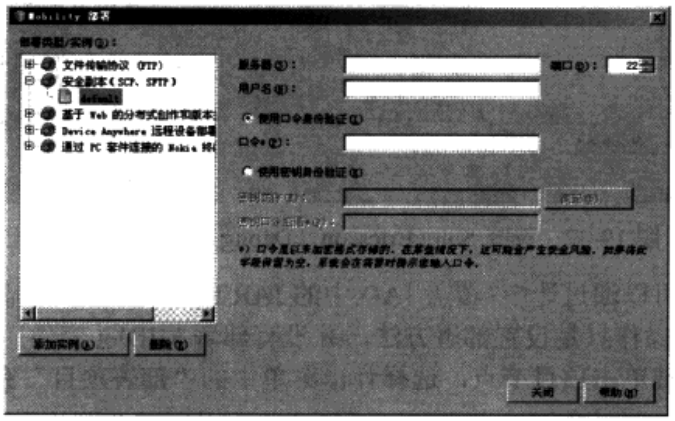


图 19-40　使用 SCP 或 SFTP 部署 MIDlet 套件

5. 部署到 Sony Dricsson 电话方法

还可以自动将 MIDlet 套件部署到 Sony Ericsson 移动电话。在“选择部署方法”下拉列表框中选择“Sony Ericsson 电话（本地连接的设备）”，如图 19-41 所示。必须在 NetBeans 中注册 Sony Ericsson（Debug On-Device）仿真器，才能使用该选项。可以在 http://developer.sonyericsson.com 网站上下载 Sony Ericsson Java ME SDK，关于 SDK 的安装和仿真平台在 NetBeans 集成开发环境中的注册在请参考本章前面的部分。

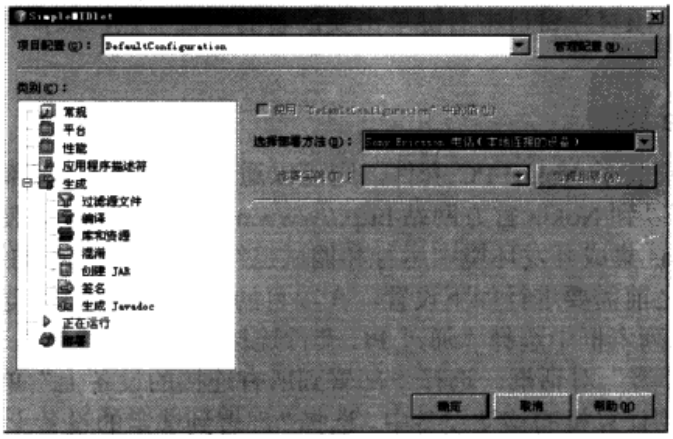


图 19-41　将 MIDlet 套件部署到 Sony Ericsson 移动电话

Download at http://www.pin5i.com/

使用 Sony Ericsson Java ME SDK 的工具和实用程序，不仅可以查看移动电话上存储的所有文件和 MIDlet，还可以在创建 MIDlet 的目标设备上运行、测试和调试这些 MIDlet。

要将 MIDlet 套件部署到移动电话上，首先将移动电话连接到电脑上。然后在 NetBeans 集成开发环境中，选择“工具”→“Java 平台”，在“平台”列表中选择 Sony Ericsson（Debug On-Device）平台，在“工具和扩展”标签中单击“打开实用程序”按钮，如图 19-42 所示。这将打开 Sony Ericsson 的“设备资源管理器”和“连接代理”窗口，此时会自动建立到移动电话设备的连接。

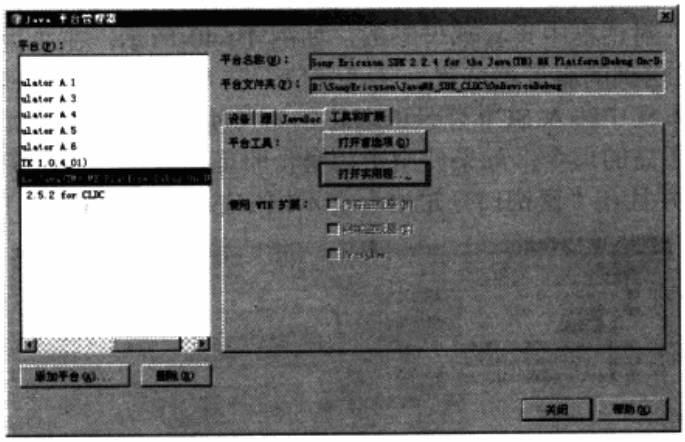


图 19-42 选择 Sony Ericsson（Debug On-Device）平台

以上几种方法都可以通过选中“覆盖 JAD 中的 JAR URL”复选框并输入特定值来定制 JAR 文件的 URL。以上的操作只是设置部署方法，并没有部署 MIDlet 套件。要部署 MIDlet 套件，在“项目”窗口中右键单击项目节点，选择弹出菜单中的“部署项目”命令。

6. 部署到 Windows Mobile 设备

使用 Sun's CLDC/MIDP 虚拟机和 ActivSync 可以将项目部署到 Windows Mobile 5 设备上。将要部署的 Windows Moble 5 设备需要安装有 Sun CLDC/MIDP VM 并且正常运行。另外，ActiveSync 已经安装到计算机上，并且已经创建设备和计算机之间的连接。

要将项目部署到 Windows Mobile 设备，在“选择部署方法”下拉列表框中选择“Windows Mobile 部署”，单击“管理部署”按钮，在打开的“Mobility 部署”对话框中，添加一个定制的实例或者使用当前设备作为默认实例。返回到属性对话框，单击“浏览”按钮，导航到 MIDP 应用程序的 JAD 和 JAR 文件要复制到设备上的位置，单击“确定”按钮。

设置完成后，在“项目”窗口中右键单击要部署的项目节点，选择弹出菜单中的“部署项目”命令。

7. 部署到 Nokia 终端

如果系统中已经安装了 Nokia PC 套件，则可以通过该套件将项目部署到 Nokia 移动设备上。Nokia PC 套件可以到 Nokia 官方网站 http://www.nokia/com 下载。使用这种部署方法，开发人员可以在 NetBeans 集成开发环境中运行和调试已经部署应用程序的设备。

在部署应用程序之前需要进行如下设置。在项目的属性对话框中，选择“部署”节点，在“选择部署方法”下拉列表框中选择“通过 PC 套件连接到 Nokia 终端”。单击“管理部署”按钮，打开“Mobility 部署”对话框。选择“部署到所有连接的设备上”单选按钮，可以将项目部署到所有已经连接到计算机的移动设备中。选择“部署到选定的设备上”单选按钮，并在“所有连接的设备”列表中选择要部署的设备，可以将项目部署到指定的设备上，如图 19-43 所示。

Download at http://www.pin5i.com/

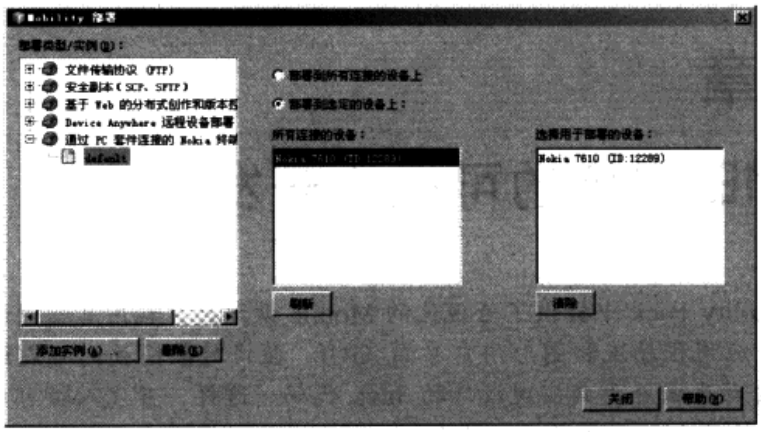


图 19-43 将 MIDlet 套件部署到 Nokia 终端

设置完成后，在“项目”窗口中右键单击要部署的项目节点，选择弹出菜单中的“部署项目”命令。

19.6 小结

本章主要介绍 Mobiltiy Pack 的安装、摩托罗拉的 Java ME 开发包的安装和在 NetBeans 中的集成、简单 Java ME 应用程序的创建和运行、NetBeans 为解决设备分裂问题提出的项目配置和预处理的概念，最后介绍了对项目各种属性的设置，通过本章的学习，读者应该掌握以下基本内容：

- Mobiltiy Pack 的安装
- 第三方开发包的安装和在 NetBeans 集成开发环境中的注册
- 理解项目配置、性能以及预处理的概念
- 仿真平台属性的设置
- 库和资源的添加
- 推注册和 API 权限的设置
- 熟悉常用的部署方法

Download at http://www.pin5i.com/

第20章

Java ME 程序的可视化开发

NetBeans Mobility Pack 中提供了可视化的 Mobile 设计器，利用其提供的强大功能可以通过可视方式来设计应用程序流和图形用户界面 GUI。在将对象拖放到流设计器和屏幕设计器后，NetBeans 将自动生成并更新实现设计的 Java 代码。这样，开发人员可以将更多的精力集中于业务逻辑上，从而大大提高了开发效率。

本章首先简要介绍简单的可视 MIDlet 的创建方法，以及可视化的 Mobile 设计器的使用方法。然后，通过开发实例具体讲解每个 MIDP 组件的使用方法，特别是对 Mobility Pack 提供的 SplashScreen、WaitScreen、FileBrowser、TableItem 等组件以及各种 SVG 组件进行了详细的介绍。

本章主要内容

- 可视 Mobile 设计器的基本构成
- 创建可视 MIDlet 的基本流程
- 利用可视 MIDlet 创建各种 MIDP 高级界面组件
- 各种组件的属性设置

本章亮点

- 可视 Mobile 设计器的使用方法
- SplashScreen、WaitScreen 和 Table 组件的使用
- SVG 组件的使用
- Mobile Game Builder 的使用

20.1 创建可视的 MIDlet

在 NetBeans 集成开发环境中利用 Mobility Pack 软件包可以实现 MIDlet 的可视化开发，本节以一个简单的示例讲解 NetBeans 中可视 MIDlet 的开发环境和开发可视 MIDlet 的基本流程。

在“项目”窗口中单击右键，选择弹出菜单中的“新建项目”（快捷键为 Ctrl + Shift + N）命令，打开“新建项目”对话框，在“类别”列表中选择 Mobile 节点，在“项目”列表中选择“Mobile 应用程序”。单击“下一步”按钮，在“项目名称”文本框中输入“VisualMobile”，并取消“创建 Hello MIDlet”复选框。



说明：如果选中“创建 Hello MIDlet”复选框，那么将自动创建下面的 Hello Visual MIDlet。这里为了给读者演示具体的可视化 MIDlet 的创建过程，因此单独进行创建。

单击“下一步”按钮，打开“选择默认平台”对话框。在“仿真器平台”下拉列表框中选择“Sun Java Wireless Tooket 2.5.2 for CLDC”，其他保持默认，单击“完成”按钮创建项目。

Download at http://www.pin5i.com/

20.1.1 创建可视化的 MIDlet

在“项目”窗口中右键单击“VisualMobile”项目节点，选择弹出菜单中的“新建”→“其他”命令，打开“新建文件”对话框，在“类别”列表中选择“MIDP”节点，在“文件类型”列表中选择“Visual MIDlet”，如图 20-1 所示。

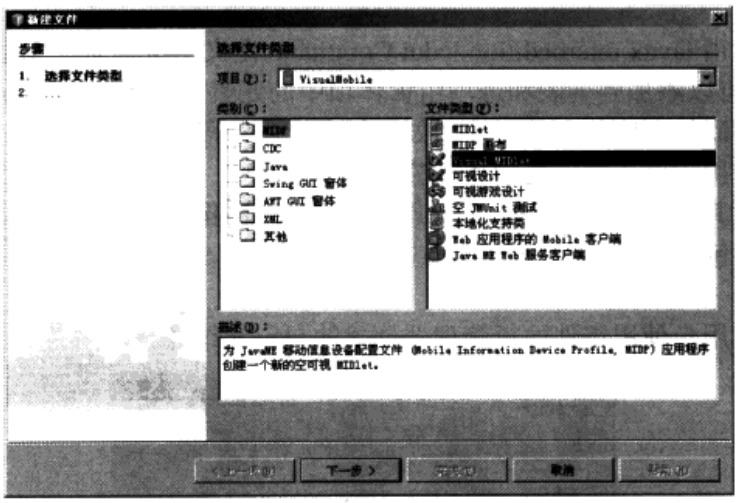


图 20-1 “新建文件”对话框

单击“下一步”按钮打开“名称和位置”对话框，如图 20-2 所示。在“MIDlet 名称”文本框中输入 MIDlet 的名字，该名字将来会显示在手机中作为程序的名字。在“MIDP 类名”文本框中输入类名。默认情况下，类名和 MIDlet 名称一致。如果把 MIDlet 名字改为中文，那么请在这里使用另外的类名。

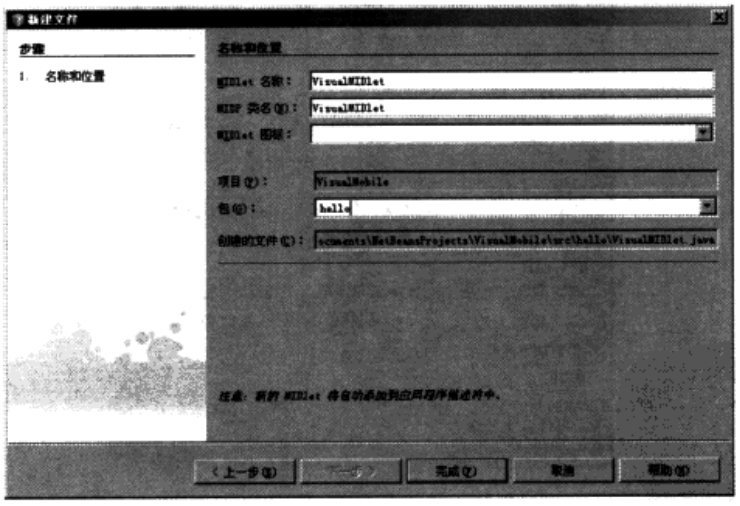


图 20-2 “名称和位置”对话框

如果项目中添加了图标，可以在“MIDlet 图标”下拉列表框中选择该图标。这里指定的图标，将来在程序安装到手机中后会作为程序的图标。在“包”文本框中输入 MIDP 类所有的包，此处输入“hello”。

单击“完成”按钮，创建可视的 MIDlet。新建的 VisualMidlet 会在可视 Mobile 设计器 VMD（Visual Mobile Designer）中打开。

Download at http://www.pin5i.com/

20.1.2 可视化 MIDlet 组件面板

用于可视化移动开发的组件面板如图 20-3 所示，其中包含可以使用可视 Mobile 设计器进行添加的所有有效 MIDP 组件的分组。默认情况下，组件面板中包含如下类别。

- 可显示项：也可以称为屏幕，包含提供基本可视显示的组件，如警报、窗体、列表以及文本框等。
- 命令：包含由 javax.microedition.lcdui.Command 定义的每种命令类型的组件。
- 元素：包含可以添加到列表和选项组窗体项中的元素，如 Choice Element 和 List Element。
- 项：包含可添加到窗体内的组件，如标尺和文本字段。
- 流：包含应用程序各种可能的执行流程。
- 资源：包含其他对象可以使用的资源，如字体、图像和滚动消息栏。
- SVG 组件：包含可以使用 SVG 图像文件的组件，如 SVG 闪屏、SVG 等待屏幕以及 SVG 菜单。
- 定制：包含用户使用组件面板管理器创建的自定义组件。

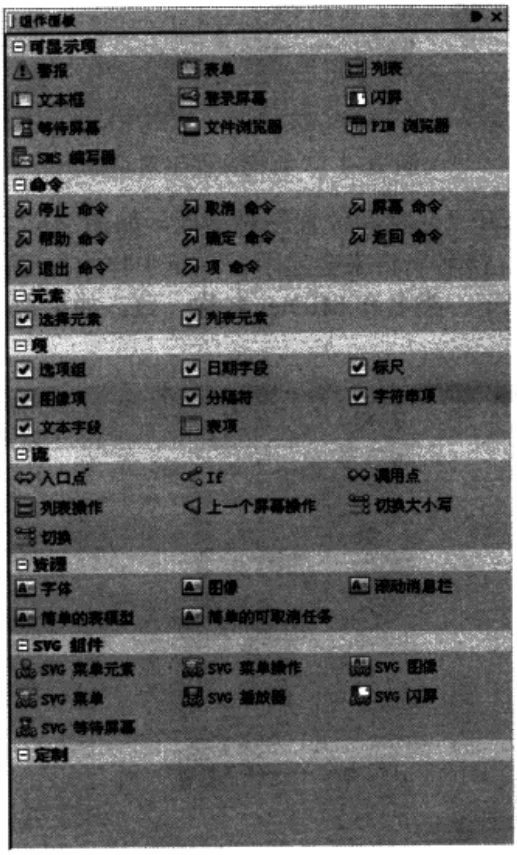


图 20-3 可视化移动开发组件面板

20.1.3 流设计器

在可视 Mobile 设计器的主工作区包含一个“流设计”标签，如图 20-4 所示。在此视图中，可以添加可显示项和其他 UI 组件，并将命令与这些组件相关联，这样可以清晰地设计出应用程序的流程。

Download at http://www.pin5i.com/

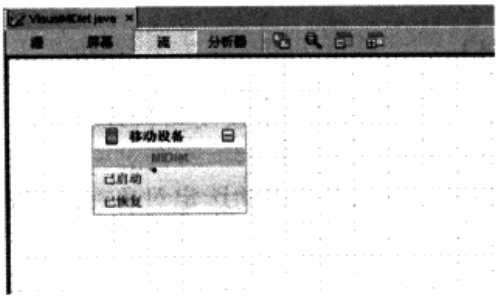


图 20-4 流设计器

所有设计都包含一个移动设备元素，此元素被用作 MIDlet 的起点。可以用鼠标将可显示项（警报、窗体、列表或文本框）从组件面板拖到“流设计”视图中，然后再将其添加到应用程序流中。

在“流设计”视图的工具栏中的按钮功能介绍如下。

- “布局”按钮：NetBeans 会自动将流设计器的组件对齐。
- “概述”按钮：显示流设计器中所有的组件的缩略图。
- “全部折叠”按钮：折叠流设计器中所有的组件，也可以单击某个组件右上方的“折叠”按钮来折叠该组件。显示网格并将屏幕按网格对齐。
- “全部展开”按钮：展开流设计器中所有的组件，也可以单击某个组件右上方的“展开”按钮来展开该组件。

20.1.4 屏幕设计器

在流设计器中，双击某个屏幕将打开“屏幕设计器”视图来编辑该屏幕，图 20-5 所示为一个 WaitScreen 组件的屏幕。

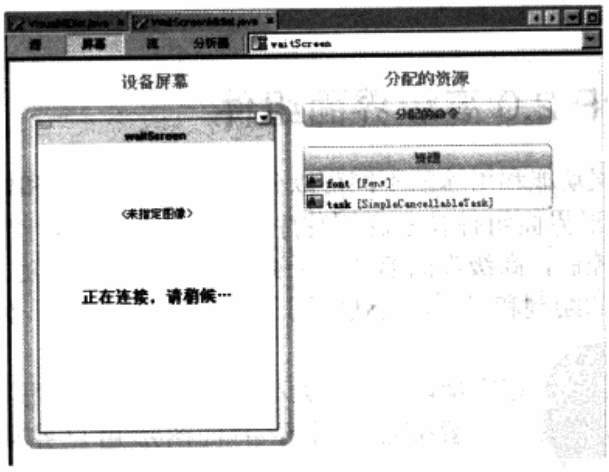


图 20-5 屏幕设计器

说明：可以按 Ctrl + Shift + Right 或 Ctrl + Shift + Left 组合键在“源”视图、“屏幕设计”视图和“流设计”视图之间切换。

在流设计器中双击某个屏幕图标后，可以在打开的“屏幕设计器”编辑该屏幕。在屏幕设计器中会显示设备的屏幕并在其右侧显示已经分配到该屏幕的资源。可以将所有有效的组件（命令、窗体项、元素、资源和定制组件等）从组件面板中拖放到设备屏幕的所需位置。

Download at http://www.pin5i.com/

20.1.5 “导航”窗口

可视 Mobile 设计器的“导航”窗口显示了使用可视设计器添加到应用程序中的所有对象，如图 20-6 所示。导航窗口以菜单树的形式显示了所有对象。可以使用菜单树在菜单中添加或删除对象，或者更改对象的顺序。导航窗口中的选定对象在“屏幕设计”视图和“流设计”视图中被突出显示。

20.1.6 “属性”窗口

“属性”窗口显示了在可视 Mobile 设计器中选定的任何对象的可编辑属性，例如，对象的实例名称、构造方法的参数以及在屏幕或对象上选择特定命令时发生的操作等。在“检查器”窗口、屏幕设计器或流设计器中选定特定的组件对象，在“属性”窗口中就会显示与之对应的属性，如图 20-7 所示。“属性”窗口中所显示的属性类型取决于选定对象的类别和类型。

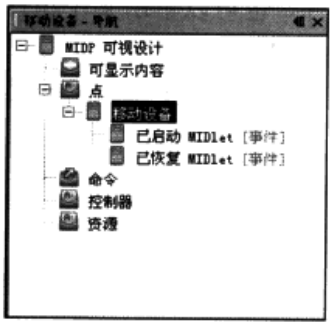


图 20-6 “导航”窗口



图 20-7 “属性”窗口

20.2 MIDP 2.0 高级界面组件

MIDP 用户界面按照功能和用途的不同可以分为高级用户界面和低级用户界面。高级用户界面主要利用已有的图形界面组件显示应用程序的内容，而具体组件的绘制由底层系统来完成，开发人员无法精确控制，高级界面通常用于商业应用程序的开发。低级用户界面可以由开发人员精确控制其具体的绘制和显示，它通常用于手机游戏的开发，低级用户界面将在下一章中介绍。

由于手机 CPU 速度、内存容量以及屏幕大小的种种限制，使用其界面显示将不同于桌面计算机，因此 MIDP 提供了一套完整的用于手机图形用户界面开发的类库，可以在 javax.microediton.lcdui 包中找到与各个图形用户界面组件相对应的接口和类。开发人员只需关注这些类的使用，而不必关心其具体实现，其实现完全依赖于各个手机厂商，这就使得开发人员开发出的应用程序可以在不同生产商的手机上运行。

MIDP 图形用户界面的显示是以屏幕为单位的，也就是说一次只能显示一个屏幕的内容，一个应用程序的显示实际上就是在各个屏幕之间切换，完成必要的操作实现其功能。在屏幕中可以加入各种组件以满足不同的需求，有些组件可以单独显示在屏幕中，而有些则必须加入到其他组件中才能显示出来，图 20-8 列出了各个组件之间的关系。

Download at http://www.pin5i.com/

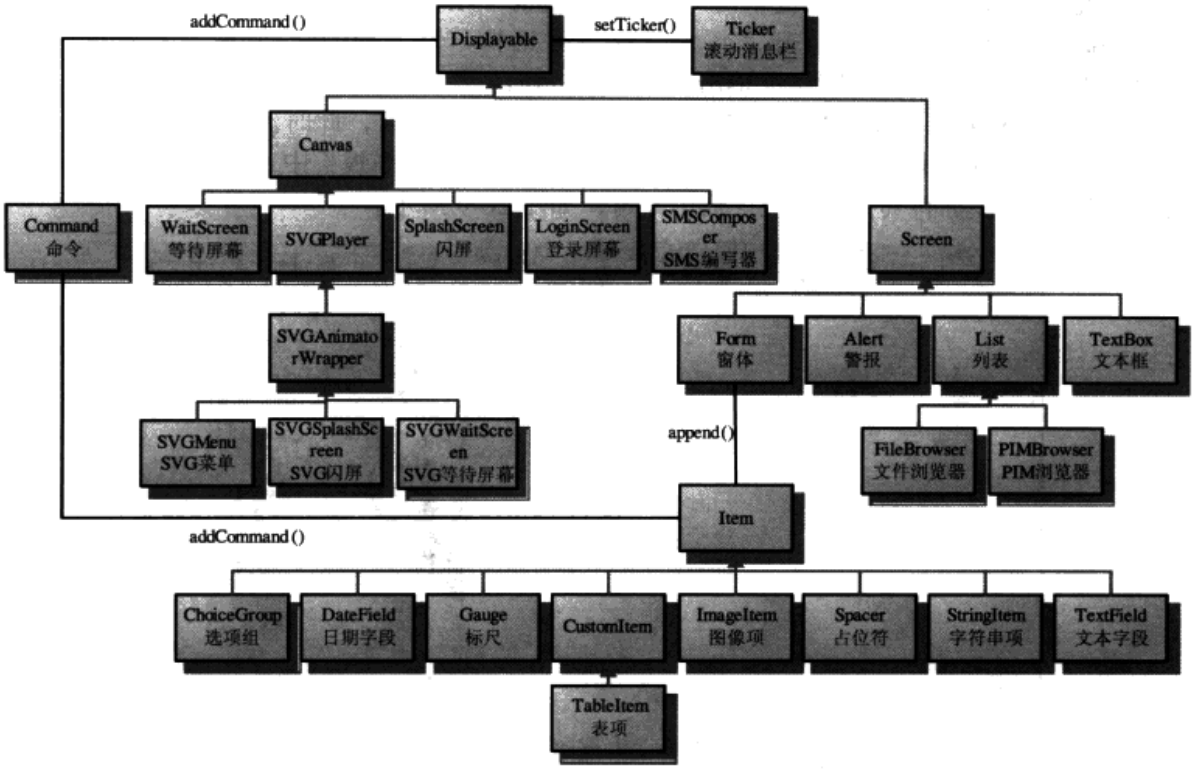


图 20-8 MIDP 2.0 高级界面组件及其关系

说明：闪屏、等待屏幕、表项等以及 SVG 相关组件为非标准 MIDP 组件，是 Mobility Pack 提供的定制组件，从图 20-8 中可以看出这些组件和 MIDP 标准组件之间的继承关系。

20.3 创建基于文本框（TextBox）的应用程序

可以利用可视 Mobile 设计器的拖放功能、GUI 环境来设计应用程序的工作流以及各个屏幕，并可以编辑各个组件属性。Mobility Pack 软件包除了提供标准的应用程序组件和命令之外，可视 Mobile 设计器还提供了用于创建表项、等待屏幕、闪屏的定制组件，并且支持可缩放矢量图形（Scalable Vector Graphics，SVG）及 SVG 动画。从本节开始，将介绍如何使用可视 Mobile 设计器进行 Java ME 应用程序的开发。

20.3.1 创建文本框应用程序

TextBox 继承自 Screen 类，可以将其单独添加到屏幕中，用户可以在 TextBox 中输入和编辑文本。

❶ 在 NetBeans 集成开发环境中，选择“文件”→“新建项目”命令，打开“新建项目”对话框。在“类别”列表中选择“Mobile”节点，在“项目”列表中选择“Mobile 应用程序”节点。

❷ 单击“下一步”按钮，打开“新建 Mobile 应用程序”对话框。在“项目名称”文本框中输入项目名称，这里为“TextBoxDemo”。取消选择“创建 Hello MIDlet”复选框。

❸ 单击“下一步”按钮，打开“选择默认平台”对话框。在“仿真器平台”下拉列表框中选择“Sun Java Wireless Tooket 2.5.2 for CLDC”，其他保持默认，单击“完成”按钮。

Download at http://www.pin5i.com/

❹ 在“项目”窗口中，右键单击“TextBoxDemo”项目节点，选择弹出菜单中的“新建”→“Visual MIDlet”命令，打开“新建文件”对话框，在“MIDlet 名称”文本框中输入“TextBoxMidlet”，其他保持默认，单击“完成”按钮。

❺ TextBoxMidlet.java 会自动在可视 Mobile 设计器中打开，此时流设计器中只有一个移动设备元素。要将文本框添加到流设计器中，只要找到“组件面板”中“可显示项”类别中的“文本框”，并用鼠标将其拖到流设计器中即可，如图 20-9 所示。

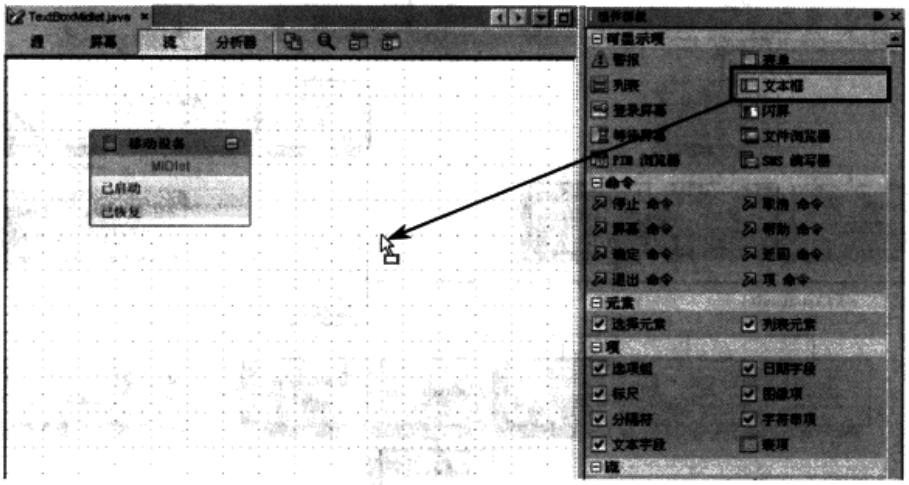


图 20-9 拖动文本框组件到流设计器中

❻ 将 MIDlet 的起点指定到此 TextBox，用鼠标单击移动设备元素上的“已启动”端口，拖动鼠标到 textBox 组件上，释放鼠标就会出现从“已启动”指向 textBox 的直线，如图 20-10 所示。

❼ 要编辑 TextBox 中的内容，在流设计器中双击 TextBox，或在 TextBox 上单击右键选择弹出菜单中的“编辑”命令。TextBox 会在屏幕设计器中打开，将鼠标放在屏幕上时，会显示闪动的红色虚线框。在虚线框内双击鼠标就可以编辑 TextBox 的内容了，如图 20-11 所示。

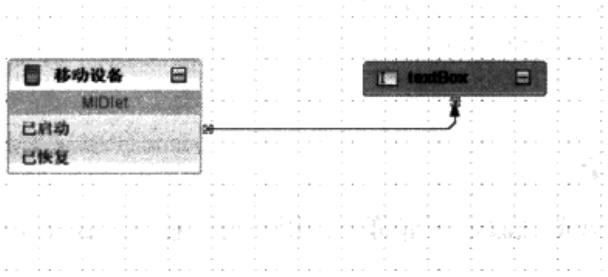


图 20-10 指定起点

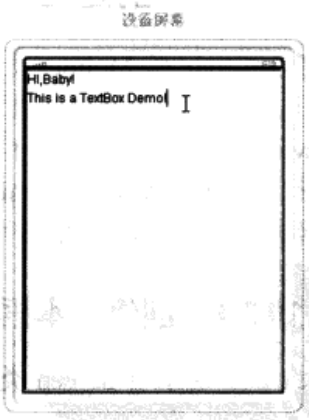


图 20-11 编辑 TextBox 的内容

❽ 按 F6 键运行主项目，在打开的模拟器窗口中启动 TextBoxMidlet。可以在模拟器的屏幕上看到文本框中的内容，如图 20-12 所示。

20.3.2 TextBox 的常用属性

在“属性”窗口中列出了所有 TextBox 的可编辑属性及其常用属性，如表 20-1 所示。

Download at http://www.pin5i.com/



图 20-12 TextBoxMidlet 运行结果

表 20-1 TextBox 的常用属性

属 性	说 明
标题	显示在屏幕上方的 TextBox 的标题
字符串	TextBox 中的文本内容
最大大小	TextBox 所能容纳的最大字符数
输入限制	TextBox 中文本的约束，单击右侧的“定制”按钮，可以在打开的“输入限制”对话框中根据需要进行定制，如图 20-13 所示。例如，选中“任意”将可以输入任意字符；选中“数字”则只能输入数字；选中“口令”复选框则输入的内容会显示为星号等其他内容
初始输入模式	用户的文本初始输入模式。设置这个属性可以在用户开始编辑文本框的内容时请求具体设备实现使用特定的输入模式。输入的属性值可以为“UCB__”加上 Unicode 字符子集（在 java.lang.Character.UnicodeBlock 中定义）的名称（例如 UCB_GREEK）。“IS__”加上输入子集（在 java.awt.im.InputSubset 定义）名称（例如 IS_SIMPLIFIED_HANZI）和 MIDP 定义的字符子集名称（MIDP_UPPERCASE_LATIN 大写字母和 MIDP_LOWERCASE_LATIN 小写字母）。例如输入属性值“UCB_HIRAGANA”应用程序会请求初始输入模式为日文。注意：设置的初始输入模式必须在移动设备支持和的情况下才有会生效
滚动消息栏	与 TextBox 关联的滚动消息栏（关于滚动消息栏将在本章后面部分介绍）
实例名称	此 TextBox 的实例名称
滞后初始化	如果选中该选项，则在第一次需要组件时初始化组件，自动生成 get_myComponent()（获取组件的实例）方法。如果未选中该选项，则在调用 MIDlet 的 startApp()方法时初始化组件。注意：如果要更改初始设置，由于可视 Mobile 设计器并不更改位于受保护块（自动生成的代码块）之外的源代码，因此必须转至该源代码并手动对其进行编辑

Download at http://www.pin5i.com/

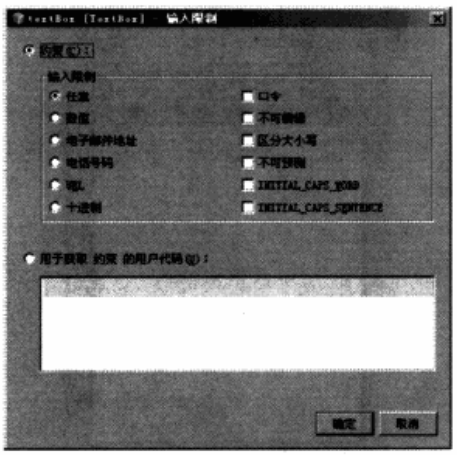


图 20-13 “输入限制”对话框



说明：对于各种组件共有的属性例如实例名称、滞后初始化等，在本章后面部分介绍其他组件时将不再进行讲解，请读者参考相关属性的说明。



20.3.3 使用命令按钮

可以使用命令按钮和用户进行交互，通过在组件上注册按钮监听器来监听按钮的动作并执行相关的代码。在“组件面板”中的“命令”类别下列表出所有 MIDP 命令按钮类型。下面将命令按钮添加到 TextBoxMidlet 中。

❶ 向流设计器中添加一个文本框，实例名称为“textBox1”。从“组件面板”中拖动一个“确定命令”到 textBox 上，其实例名称为“okCommand”，如图 20-14 所示。用同样的方法，向 textBox 添加一个“退出命令”，实例名称为“exitCommand”。

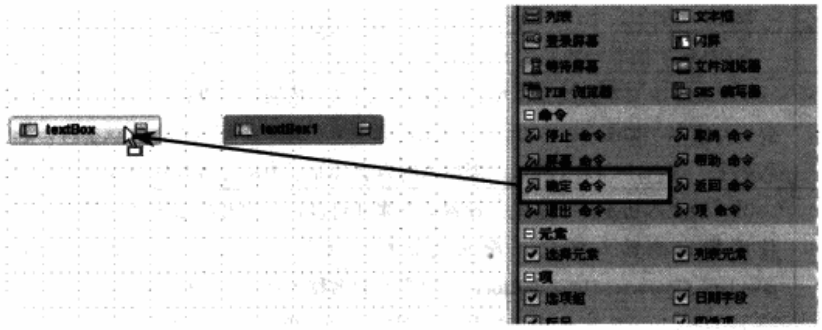


图 20-14 添加确定命令按钮

❷ 双击 textBox，打开 textBox 的屏幕设计器，在设置屏幕的右侧列出已为 textBox 分配的命令，此处只有 okCommand，用鼠标单击将其选中，选中的组件周围会出现橙色的实线，如图 20-15 所示。

❸ 在“属性”窗口中修改 okCommand 的“操作”属性，在打开的“操作”对话框中编辑命令按钮的操作，如图 20-16 所示。可以选择“切换为可显示”单选按钮，并选择然后转到的屏幕，实现屏幕之间的导航。选择“调用”单选按钮可以执行已定义的流动作。选择“转至上一个可显示内容”单选按钮可以返回到前面显示的屏幕。选择“退出应用程序”单选按钮则会调用 notifyDestroyed()方法退出应用程序。如果选择“不执行任何操作”单选按钮则单击命令按钮时将不执行任何切换。此处选择切换到 textBox1。

Download at http://www.pin5i.com/

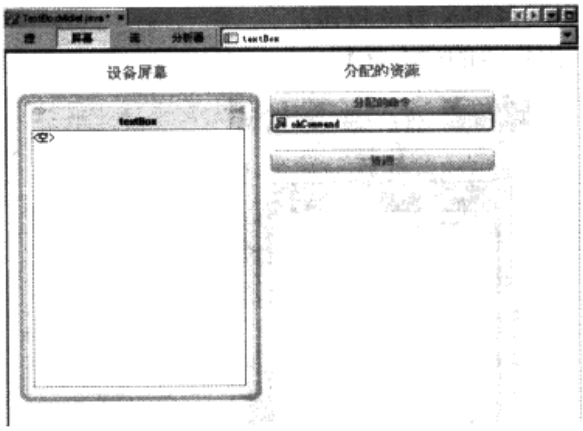


图 20-15 分配的命令

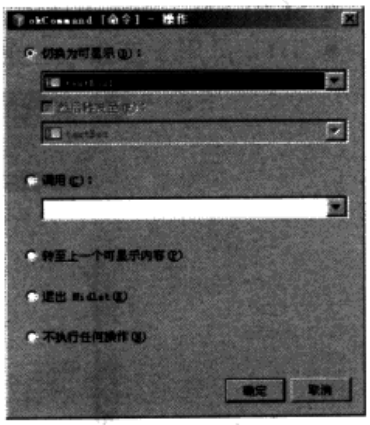


图 20-16 “操作”对话框

说明：在流设计器中双击某个转至源的直线可以打开源代码编辑器并转到相应操作的代码位置。

❹ 在屏幕设计器工具栏上的“编辑的屏幕”下拉列表框中选择 textBox1，textBox1 就会在屏幕设计器中打开，编辑 textBox1 的内容为“This is textBox1!”。

❺ 切换到流设计视图，连接移动设备元素的“已启动”端口到 textBox 上。连接 textBox 组件上的“okCommand”到 textBox1 上。连接 textBox1 上的“exitCommand”到移动设备元素上，最终流设计如图 20-17 所示。

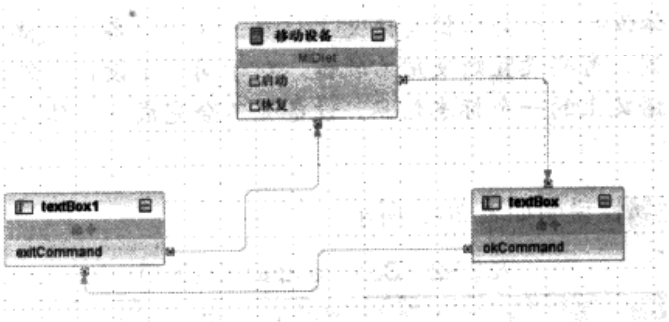


图 20-17 流设计结果

❻ 按 F6 键运行主项目，在打开的模拟器窗口中启动 TextBoxMidlet。可以使用命令按钮在两个文本框之间导航和执行相应代码，如图 20-18 所示。

选中 exitCommand，可以在“属性”窗口中查看其属性。其中，命令按钮的类型属性用于指明使用按钮的意图。例如，如果在应用程序中要实现“返回”功能，则应该指定命令按钮的类型为 BACK。在设备具体实现时，如果该设备有标准的执行“返回”操作的软键与之相对应，那么就会把这个命令按钮和该软件关联起来。以下类型为预定义的命令按钮。

- BACK：返回给用户逻辑上的前一个屏幕的导航按钮。
- CANCEL：对当前屏幕对话框一个标准否定回答的命令按钮。
- EXIT：用于退出应用程序的命令按钮。
- HELP：指定一个联机帮助的命令按钮。
- ITEM：应用程序使用此按钮类型指示具体设备实现此命令按钮适应于特定的屏幕项目或列表或选项组中的元素。
- OK：对当前屏幕对话框一个标准肯定回答的命令按钮。

Download at http://www.pin5i.com/

- SCREEN：指定附属于当前屏幕的应用程序自定义命令按钮。
- STOP：用于停止一些正在运行的进程、操作等的命令按钮。



图 20-18　使用命令按钮的 TextBoxMidlet

说明：具体设备实现会根据其语义上信息来保持设备的风格，有些设备使用左软键来作为返回功能键，有些设置则使用右软键。对于同一个设备，风格应该是一致的。命令按钮的类型只是语义上的一个标志信息，设备并不会完成其具体操作，其操作代码必须在应用程序中编写。

Command 的其他属性说明如表 20-2 所示。

表 20-2　Command 的属性说明

属　性	说　明
操作	单击按钮时执行的操作，可以单击其后的“定制”按钮 打开“操作”对话框进行编辑
操作源	当前操作的操作源，也就是当前操作是由哪个命令按钮触发的
标签	命令按钮上显示的标签
长标签	命令按钮的长标签。如果屏幕的空间能够显示长标签则显示长标签的内容，否则会显示标签的内容
优先级	命令按钮的优先级。优先级表示某个命令按钮在当前屏幕的重要程度，其值为一个整型值，数值越小优先级越高
类型	命令按钮的类型。应该使用命令按钮类型指明

20.4　创建闪屏（SplashScreen）

闪屏可以增强应用程序的视觉效果。一般在应用程序启动时使用闪屏，用其显示 logo 或商标信息。闪屏将给用户留下应用程序的第一印象。

使用可视 Mobile 设计器的闪屏组件可以很容易地设计一个闪屏。

❶ 在 NetBeans 集成开发环境中，新建一个“Mobile 应用程序”项目，项目名称为

Download at http://www.pin5i.com/

“SplashScreenDemo”，不创建 Hello MIDlet，仿真平台选择“Sun Java Wireless Toolkit 2.5.2 for CLDC”。在“SplashScreenDemo”项目中新建一个 Visual MIDlet，名称为“SplashScreenMidlet”。

❷ 在可视 Mobile 设计器中，用鼠标将组件面板中的 SplashScreen 拖动到 SplashScreenMidlet 的流设计器中，选中 spashScreen 组件，在属性窗口中修改其“文本”属性为“请等待 6 秒中或按任意键关闭闪屏”。修改“超时”属性的值为“6000”，或者单击后面的定制按钮在打开的“超时”对话框中设置超时时间，可以直接在“超时”文本框中输入数值，也可以在“用于获取整型的用户代码”文本框中输入代码来获取超时时间，如图 20-19 所示。

❸ 拖动一个 TextBox 到流设计器中，再拖动一个退出命令到 textBox 组件中。双击 textBox 组件将其在屏幕设计器中打开，选中 textBox 屏幕并双击，修改其文本内容为“闪屏完毕，请单击退出按钮退出应用程序”。

❹ 为了增强闪屏的显示效果，可以向其中添加图像。将预先准备好的图像文件复制到项目的 src 文件夹中，在“导航”窗口中右键单击“资源”节点，选择弹出菜单中的“新建/添加”→“图像”命令，如图 20-20 所示。

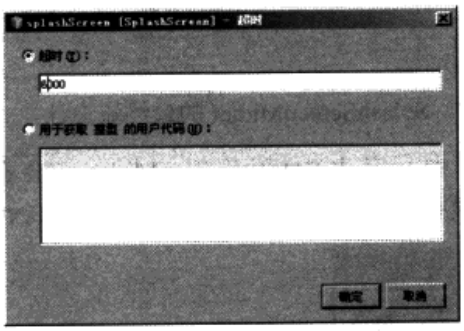


图 20-19 “超时”对话框

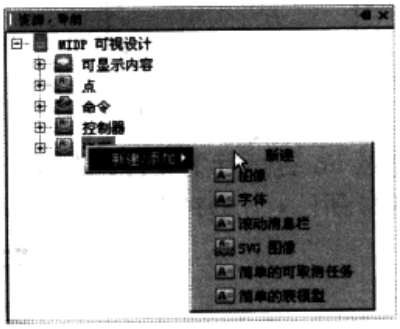


图 20-20 添加图像

❺ 选中 image 节点，在“属性”窗口中单击“资源路径”属性后的“定制”按钮，打开“资源路径”对话框，如图 20-21 所示。可以在“图像路径”下拉列表框中选择刚才添加的图像文件，可以在下方预览图像以及图像大小和文件大小等信息，也可以单击“浏览”按钮，将外部图像文件导入到项目文件夹中。同时，还可以在“用于获取图像资源路径的用户代码”下面的文本框中输入代码来获取图像资源。单击“确定”按钮，完成图像文件的选择。

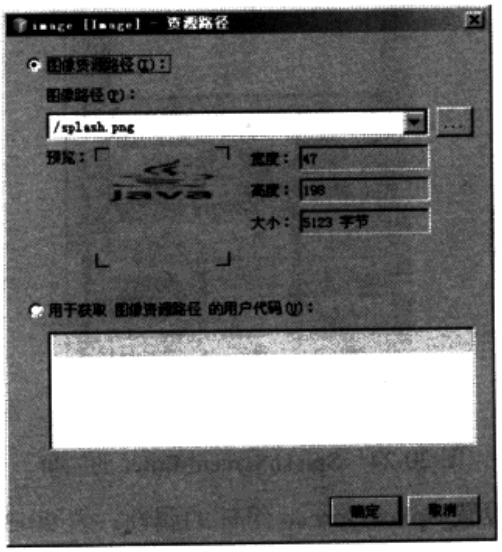


图 20-21 “资源路径”对话框

Download at http://www.pin5i.com/

❻ 在流设计器中选中 splashScreen 组件，在“属性”窗口中修改“图像”属性，选择其后下拉列表框中的 image。双击 splashScreen，可以在屏幕设计器中看到 splashScreen 的屏幕如图 20-22 所示。

❼ 返回到流设计器中，用鼠标连接移动设备元素上的“已启动”端口到 splashScreen 组件上，连接 splashScreen 上的 DISMISS_COMMAND 端口到 textBox 上，最后连接 textBox 上的“exitCommand”到移动设备上，最终结果如图 20-23 所示。



图 20-22 闪屏屏幕

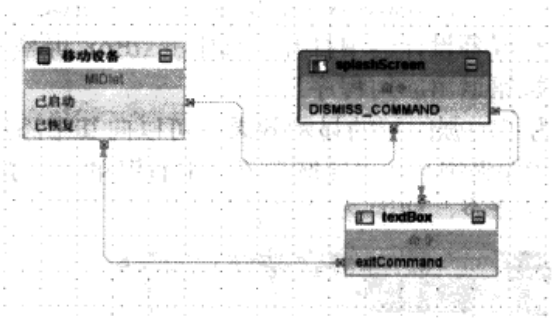


图 20-23 SplashScreenMidlet 的最终流设计

❽ 按 F6 键运行主项目，在打开的模拟器窗口中，启动 SplashScreenMidlet，会看到闪屏如图 20-24 所示。6 秒钟之后或者用鼠标单击一下模拟器上的按键，屏幕会切换到 TextBox。



图 20-24 SplashScreenMidlet 的闪屏

可以在“属性”窗口中设置 SplashScreen 组件的属性，表 20-3 列出了 SplashScreen 组件的属性。

Download at http://www.pin5i.com/

表 20-3 SplashScreen 的属性说明

属 性	说 明
超时	闪屏显示的时间，单位为毫秒
允许超时中断	是否允许超时中断。如果为 true，则用户可以在闪屏超时中断，执行下一步操作，否则用户只能等待超时时间完毕
关闭操作	当超时时间到或用户中断时所执行的操作
文本	闪屏窗口中显示的文本内容
图像	闪屏窗口中显示的图像
文本字体	闪屏窗口中显示文本的字体格式
全屏	是否全屏显示
标题	闪屏窗口的标题，如果全屏属性设置为 true，将不会显示标题

20.5 使用滚动消息栏（Ticker）

滚动消息栏用于显示提示信息或广告等，在滚动消息栏中会连续不断地显示横穿屏幕的文本，但其速度和方向由具体设备决定。任何一个 Displayable 对象都可以通过使用 setTicker()方法设置滚动消息栏。

下面介绍如何在可视 Mobile 设计器中使用滚动消息栏。

❶ 新建一个 Mobile 应用程序项目，项目名称为“TickerDemo”，不创建 Hello MIDlet，仿真平台选择“Sun Java Wireless Toolkit 2.5.2 for CLDC”。在 TickerDemo 中新建一个 Visual MIDlet，名称为“TickerMidlet”。

❷ 用鼠标向流设计器中施放一个 TextBox，再施动一个退出命令到 textBox 组件中。将“组件面板”中资源类别下的 Ticker 施动到流设计器的 textBox 中。也可以双击 textBox 组件将其在屏幕设计器中打开，单击 textBox 屏幕右上方的下三角按钮，然后单击“指定新的滚动消息栏”按钮来添加 Ticker，如图 20-25 所示。

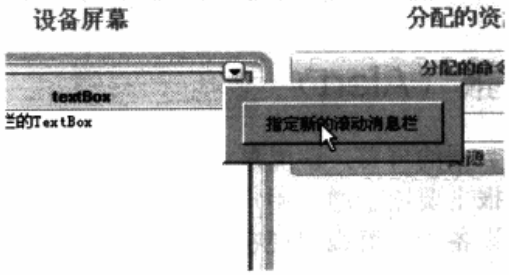


图 20-25 添加 Ticker

❸ 这时会发现“导航”窗口中的“资源”节点下出现了 ticker 节点，将其选中并在“属性”窗口中设置其“字符串”属性为“这里是滚动消息栏”。

❹ 在流设计器中，右键单击 textBox 组件，选择弹出菜单中的“编辑”命令，将在屏幕设计器中打开 textBox，修改其文本内容为“带滚动消息栏的 TextBox”。

❺ 连接移动设备元素上的“已启动”端口到 textBox 上，再连接 textBox 上的“exitCommand”到移动设备元素上。

❻ 按 F6 运行主项目，在模拟器中启动 TickerMidlet，会看到 TextBox 上方的滚动消息栏，如图 20-26 所示。

Download at http://www.pin5i.com/



图 20-26 滚动消息栏

滚动消息栏的使用比较简单，表 20-4 列出了其属性。

表 20-4 Ticker 的属性说明

属 性	说 明
字符串	滚动消息栏中滚动的消息文本

20.6 创建警报（Alert）

警报是一个向用户显示数据的屏幕，该屏幕在进入一个屏幕之前会等待一定时间。警报可以包含文本字符和图像。警报主要用于通知用户错误和其他异常条件。

警报可以有多种类型。设备会在警报出现在用户面前时根据警报的种类播放适当的声音。下面举例介绍如何在 NetBeans 集成开发环境中使用警报。

❶ 新建一个“Mobile 应用程序”项目，项目名称为“AlertDemo”，不创建 Hello MIDlet，仿真平台选择“Sun Java Wireless Toolkit 2.5.2 for CLDC”。在“AlertDemo”项目中新建一个 Visual MIDlet，名称为“AlertMidlet”。

❷ 用鼠标从“组件面板”中拖动一个 TextBox 到流设计器中，再拖动一个 Alert 到流设计器中，向 textBox 中添加一个退出命令按钮，再向 textBox 中添加 5 个 Screen Command 命令按钮，命令按钮的实例名称和标签分别为 infoCmd 和 INFO、errorCmd 和 ERROR、warnCmd 和 WARNING、alarmCmd 和 ALARM、confCmd 和 CONFIRMATION。

❸ 双击 textBox 将其在屏幕设计器中打开，修改其文本内容为“选择命令按钮，显示相应类型的警报！”。

❹ 在屏幕设计器工具栏中的“编辑的屏幕”下拉列表框中选择 alart，修改 alart 的文本内

Download at http://www.pin5i.com/

容为“这里是警报信息”，在“属性”窗口中单击“超时”属性后的“定制”按钮，在打开的“超时”对话框中修改超时的值为 3000，如图 20-27 所示。

❺ 下面为警报添加图像。在“属性”窗口中，单击“图像”属性后的“定制”按钮，打开“图像”对话框。单击“添加”按钮，这里在“资源”列表中会添加一项“image1”，单击“图像路径”右侧的浏览按钮，选择图像文件，文件会被自动复制到项目的 src 文件夹中，如图 20-28 所示。

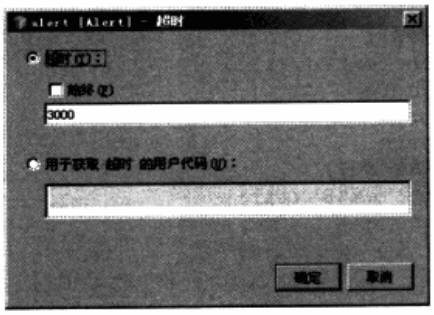


图 20-27 “超时”对话框

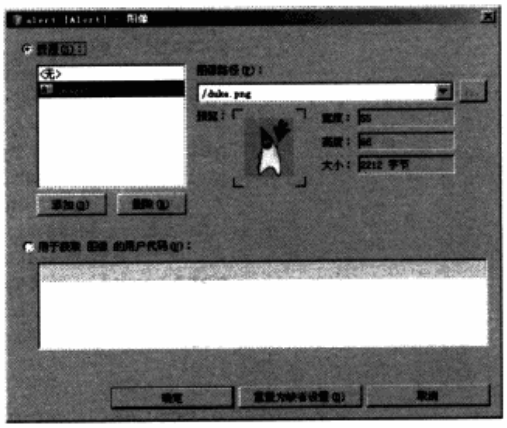


图 20-28 “图像”对话框

❻ 返回到流设计视图，用鼠标连接移动设备元素上的“已启动”端口到 textBox 上，分别连接 infoCmd、errorCmd、warnCdm、alarmCmd 和 confCmd 命令的端口到 alart 上，连接 exitCommand 命令端口到移动设备上。此时流设计器显得比较零乱，单击流设计器工具栏上的“重新对齐屏幕”按钮，流设计器的最终结果如图 20-29 所示。

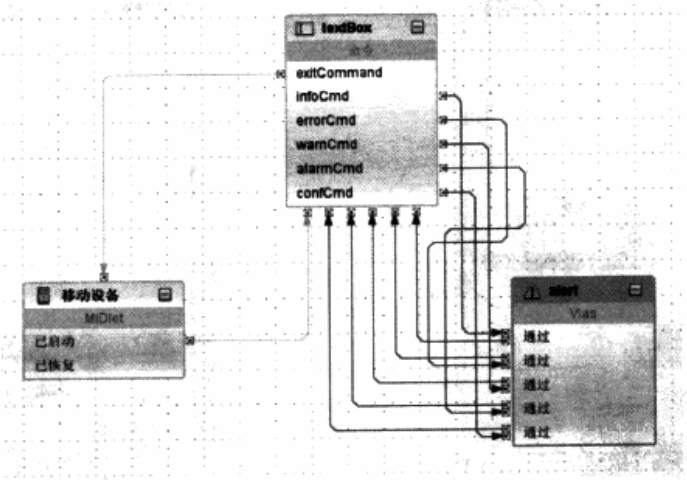


图 20-29 AlertMidlet 流设计结果

❼ 转换源视图中，修改 commandAction()方法的代码如下所示：

```
public void commandAction(Command command, Displayable displayable) {
    // write pre-action user code here
    if (displayable == textBox) {
        if (command == alarmCmd) {
            // write pre-action user code here
            getAlert().setType(AlertType.ALARM);
            switchDisplayable(getAlert(), getTextBox());
            // write post-action user code here
        } else if (command == confCmd) {
```

Download at http://www.pin5i.com/


```
                // write pre-action user code here
                getAlert().setType(AlertType.CONFIRMATION);
                switchDisplayable(getAlert(), getTextBox());
                // write post-action user code here
            } else if (command == errorCmd) {
                // write pre-action user code here
                getAlert().setType(AlertType.ERROR);
                switchDisplayable(getAlert(), getTextBox());
                // write post-action user code here
            } else if (command == exitCommand) {
                // write pre-action user code here
                exitMIDlet();
                // write post-action user code here
            } else if (command == infoCmd) {
                // write pre-action user code here
                getAlert().setType(AlertType.INFO);
                switchDisplayable(getAlert(), getTextBox());
                // write post-action user code here
            } else if (command == warnCmd) {
                // write pre-action user code here
                getAlert().setType(AlertType.WARNING);
                switchDisplayable(getAlert(), getTextBox());
                // write post-action user code here
            }
        }
        // write post-action user code here
    }
```

❽ 按 F6 键运行主项目，在模拟器中启动 AlertMidlet，在屏幕菜单中选择命令按钮，如图 20-30 所示。相应类型的警报就会显示在屏幕上，如图 20-31 所示。

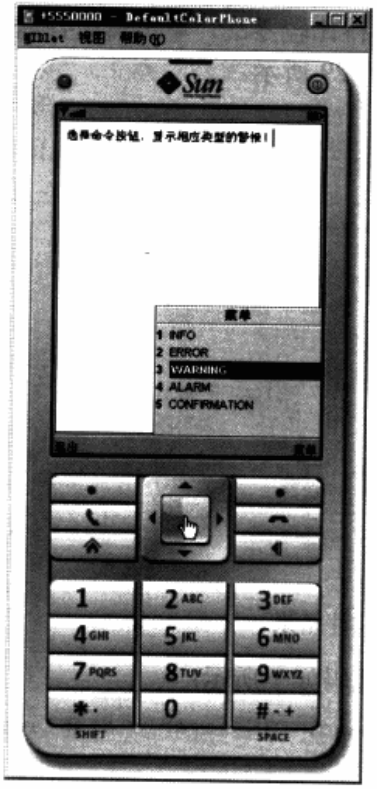


图 20-30 选择命令按钮

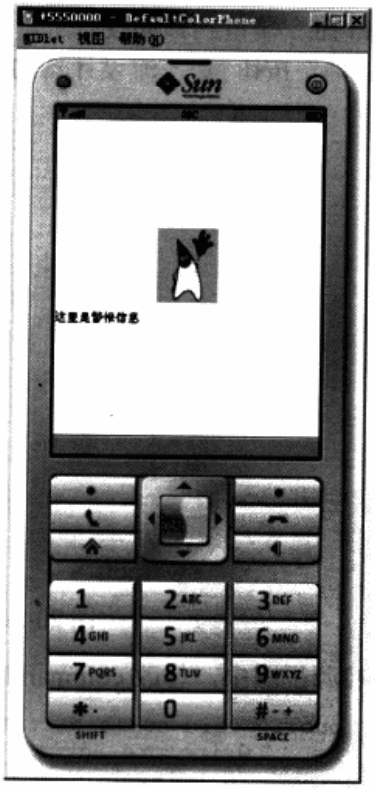


图 20-31 警报显示效果

Alert 组件的属性见表 20-5。

Download at http://www.pin5i.com/

表 20-5 Alert 组件的属性说明

属性	说明
标题	警报屏幕的标题
字符串	警报屏幕中显示的消息文本
超时	警报屏幕显示的时间，单位为毫秒
类型	警报屏幕的类型，有如下几种。 INFO：提供用户非危险信息 ERROR：警报用户错误操作的提示 WARNING：警告用户潜在的危险操作的提示 ALARM：警报用户其先前请求需要得到通知的事件的提示 CONFIRMATION：确定用户行为的提示
图像	警报屏幕中使用的图像
使用指示器	如果设置为 true，可以将 Guage 组件添加到警报屏幕中（Guage 组件将在本章后面部分介绍）

20.7 创建等待屏幕（WaitScreen）

Java ME 应用程序为了能够正常工作经常需要网络连接。然而，无线网络相对比较慢。一个等待网络连接或者数据传输的后台任务可能中止应用程序，这样可能让用户感觉到冻结的屏幕是否意味着要无限期地凝视着不动的屏幕或者要结束应用程序稍后重试。

要解决这个问题可以向屏幕中添加等待屏幕。等待屏幕允许用户执行阻塞的后台任务（例如，连接到网络），并且显示任务的进度或在屏幕上显示信息来通知用户应用程序依然在运行。

为等待屏幕组件创建后台任务有两种方法。可以创建类并实现 CancellableTask 接口或者使用 SimpleCancellableTask 资源。可以使用 setTask()方法为等待屏幕分配后台任务。当后台任务完成时等待屏幕组件会自动切换到一个不同的 Displayable 对象。如果任务执行成功，则等待屏幕切换到由 setNextDisplayable()方法提供的显示对象。如果任务执行失败，则等待屏幕切换到由 setFailureDisplayable()方法提供的显示对象，如果没有设置默认失败可显示对象，那么等待屏幕会切换到由 setNextDisplayable()方法指定的显示对象。如果没有指定下一个可显示对象，则等待屏幕会切换回先前显示的屏幕。

下面使用可视 Mobile 设计器创建带有等待屏幕的应用程序。

❶ 在 NetBeans 集成开发环境中，新建一个“Mobile 应用程序”项目，项目名称为“WaitScreenDemo”，不创建 Hello MIDlet，仿真平台选择“Sun Java Wireless Toolkit 2.5.2 for CLDC”。在“WaitScreenDemo”项目中新建一个 Visual MIDlet，名称为“WaitScreenMidlet”。

❷ 从组件面板中拖动两个 TextBox 组件、一个 WaitScreen 组件和一个 Alert 组件。

❸ 修改 textBox 和标题属性为“请输入地址”，修改 textBox1 的标题属性为“服务器信息”。修改 alert 的字符串属性为“网络连接失败”。修改 waitScreen 的文本属性为“正在连接，请稍候...”，在“文字字体”属性后的下拉列表框中选择“新的字体资源”。在导航窗口中选中 font，在属性窗口中修改“类型”属性为“定制”，然后修改“大小”属性为“大”，修改“样式”属性为“粗体”。

❹ 向 textBox 中添加一个“确定”按钮，向 textBox1 中添加一个 Back Command 和退出命令按钮。在流设计视图中，用鼠标连接各个组件，其最终流设计如图 20-32 所示。

Download at http://www.pin5i.com/

❺ 在导航窗口中，选中 task，这是为等待屏幕自动创建后台任务，每个任务都是一个 SimpleCancellableTask 实例。在属性窗口中，单击“Executable code”属性后面的定制按钮，在弹出的“Executable code”对话框中，单击“转至源”按钮转到源代码视图。

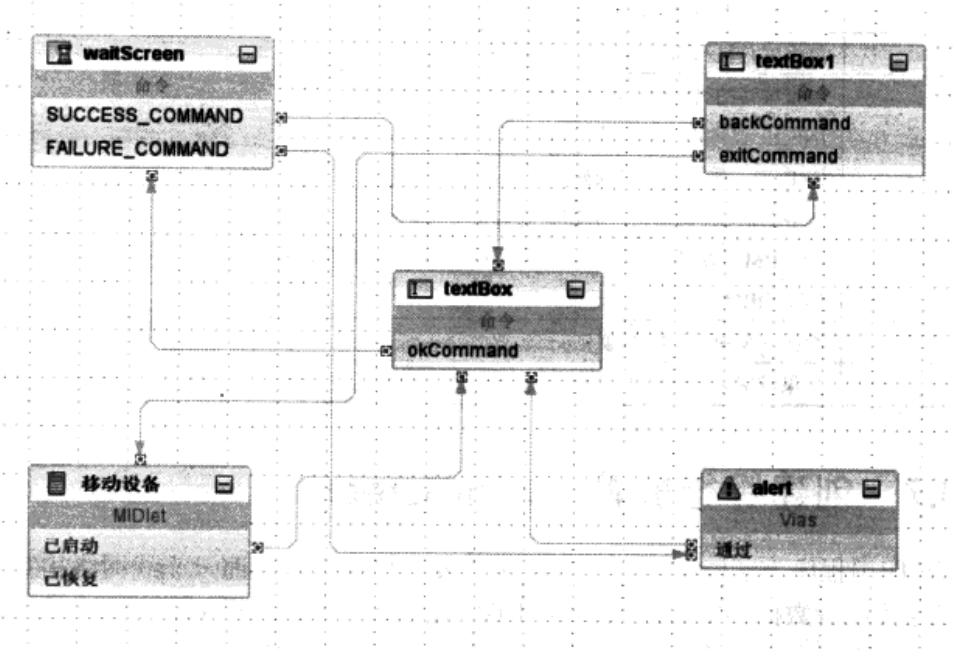


图 20-32 WaitScreenMidlet 流设计结果

❻ 修改 execute 方法的代码如下所示：

```
private void execute() throws IOException {

        String url = getTextBox().getString();
        HttpConnection conn = (HttpConnection) Connector.open(url);
        StringBuffer sb = new StringBuffer();
        sb.append("主机: "+conn.getHost());
        sb.append("\nIP 地址: "+conn.getURL());
        sb.append("\n 端口: "+conn.getPort());
        sb.append("\n 协议: "+conn.getProtocol());
        sb.append("\n 响应代码:"+conn.getResponseCode());
        sb.append("\n 响应消息: "+conn.getResponseMessage());
        sb.append("\n 请求方法: "+conn.getRequestMethod());
        sb.append("\n 查询字符串: "+conn.getQuery());
        getTextBox2().setString(sb.toString());
        conn.close();
    }
```

❼ 应用程序的设计过程至此完毕，按 F6 键运行主项目，在打开的模拟器中启动 WaitScreenMidlet。在“请输入地址”文本框中输入要连接的 IP 地址，如图 20-33 所示。按下“确定”键后会出现“是否要使用‘网络通话’”的提示，如图 20-34 所示。

❽ 按下“是”命令按钮，此时等待屏幕会显示在用户面前，如图 20-35 所示。等待屏幕所执行的任务成功时会出现如图 20-36 所示的屏幕。如果任务执行失败（例如，输入的地址有误），则会出现警报并返回到输入地址的文本框。

Download at http://www.pin5i.com/

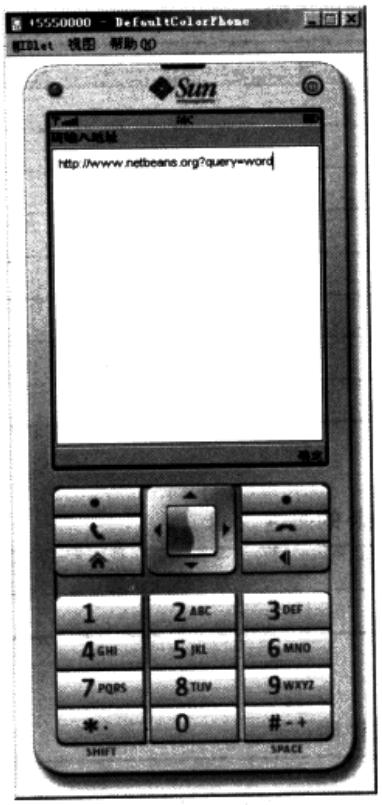


图 20-33 请输入地址



图 20-34 是否要使用“网络通话”



图 20-35 等待屏幕效果

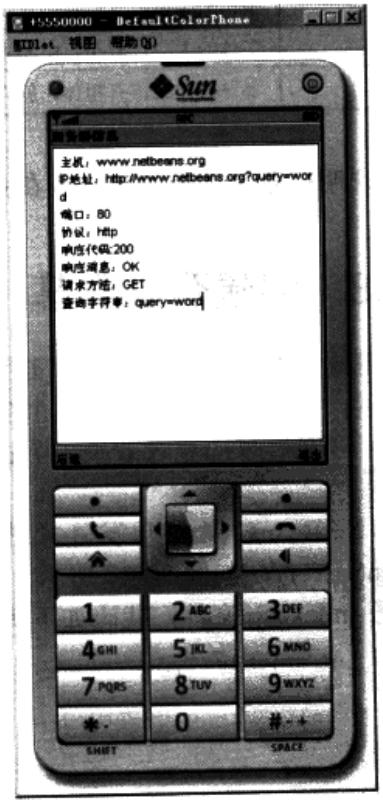


图 20-36 任务成功后显示效果

和其他组件一样，等待屏幕也有自己的属性，等待屏幕的属性如表 20-6 所示。

Download at http://www.pin5i.com/

表 20-6 WaitScreen 的属性说明

属 性	说 明
任务	等待屏幕所执行的任务
成功操作	任务成功时的操作
失败操作	任务失败时的操作
文本	等待屏幕中显示的文本
图像	等待屏幕中显示的图像
文本字体	等待屏幕中显示的文本字体格式
全屏	是否全屏显示
标题	等待屏幕的标题

SimpleCancellableTask 的属性如表 20-7 所示。

表 20-7 SimplCancellableTask 的属性说明

属 性	说 明
可执行的方法主体	SimpleCancellableTask 所执行的方法

Font 的属性如表 20-8 所示。

表 20-8 Font 的属性说明

属 性	说 明
字样	字体的字样，有系统、等宽和成比例三种
类型	字体的类型，有默认、系统和定制三种
大小	字体的大小，有小、中和大三种
样式	字体的样式，有粗体、斜体和下画线三种



说明：等待屏幕组件只有在支持 MIDP 2.0 的应用程序中才能使用。

20.8 创建登录屏幕（LoginScreen）

登录屏幕定制组件提供一个包含实用的用户接口，界面中包含用户名、密码和登录按钮这样的标准元素。开发人员可以使用这个组件创建一个登录界面来访问受保护的资源。

为了显示登录屏幕的登录效果，首先创建一个 Web 应用程序，用于验证客户端传入的用户名和密码。

❶ 在 NetBeans 集成开发环境中新建一个“Web 应用程序”项目，项目名称为“LoginScreenServer”，如图 20-37 所示。

❷ 修改 index.jsp 文件，代码如下所示：

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
    final  String USERNAME = "FeiDongdong";
    final  String PWD = "405103010404";
    String username = request.getParameter("username");
    String pwd = request.getParameter("pwd");
    if(username.equals(USERNAME) && pwd.equals(PWD)){
       response.sendError(HttpServletResponse.SC_OK);
    }else{
       response.sendError(HttpServletResponse.SC_NOT_ACCEPTABLE);
```

Download at http://www.pin5i.com/


```
    }
%>
```

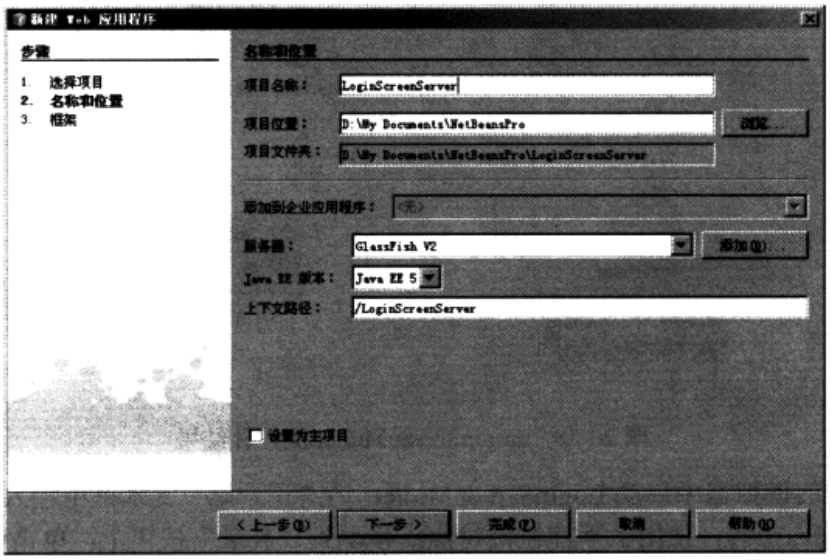


图 20-37 新建 Web 应用程序

下面使用可视 Mobile 设计器创建带有登录屏幕的应用程序。

❶ 在 NetBeans 集成开发环境中新建一个“Mobile 应用程序”项目，项目名称为“LoginScreenDemo”，不创建 Hello MIDlet，仿真平台选择“Sun Java Wireless Toolkit 2.5.2 for CLDC”。在“LoginScreenDemo”项目中新建一个 Visual MIDlet，名称为“LoginScreenMidlet”。

❷ 从组件面板中拖动一个 LoginScreen 组件、一个 WaitScreen 组件和一个 TextBox 组件到流设计器中。拖动一个退出命令按钮到 LoginScreen 组件上，拖动一个 Back Command 按钮到 TextBox 组件上。

❸ 在属性窗口中修改 LoginScreen 组件“‘登录’按钮的文本”属性的值为“登录”，修改“口令标签”属性的值为“密码”，修改“使用‘登录’按钮”属性的值为“True”，修改“用户名标签”属性的值为“用户名”。

❹ 修改 waitScreen 组件中 task 的 execute 方法的代码如下所示：

```
    public void execute() throws Exception {
    // write task-execution user code here
    boolean isLogin = false;
    String url = "http://localhost:8080/LoginScreenServer/index.jsp?username="
         + getLoginScreen().getUsername() + "&pwd=" +
getLoginScreen().getPassword();
    HttpConnection conn = (HttpConnection) Connector.open(url);
    if(conn.getResponseCode()==HttpConnection.HTTP_OK){
        isLogin = true;
    }
    if(isLogin){
       getTextBox().setString("登录成功!");
    }else{
       getTextBox().setString("登录失败! ");
    }
    }
```

❺ 在流设计视图中，用鼠标连接各个组件，其最终流设计如图 20-38 所示。

❻ 在项目窗口中，右键单击 LoginScreenServer 项目节点，在弹出的菜单中选择“运行”命令。待 LoginScreenServer 成功运行后，使用同样的方法运行项目 LoginScreenDemo。

Download at http://www.pin5i.com/

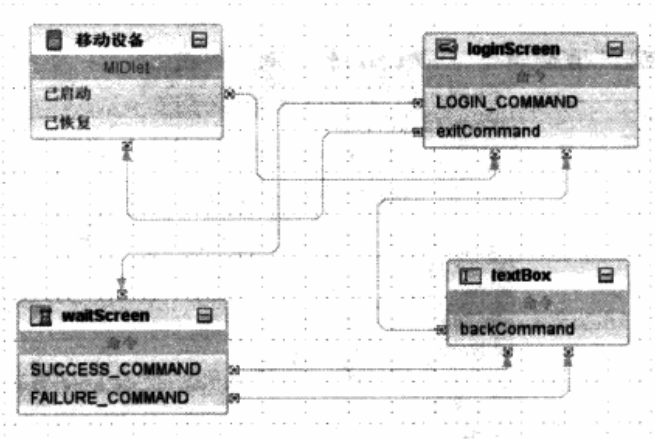


图 20-38 LoginScreenMidlet 流设计结果

❼ 在打开的模拟器中启动 LoginScreenMidlet。在“用户名”文本框中输入用户名，在“密码”文本框中输入密码，使用上下控制键将焦点移动到“登录”按钮上，单击确定按钮，如图 20-39 所示。

❽ 屏幕会出现“是否要使用‘网络通话’”的提示，按下“是”命令按钮，此时等待屏幕会显示在用户面前。如果用户名和密码都正确，会出现如图 20-40 所示的屏幕。即等待屏幕所执行的任务成功时会出现如图 20-40 所示的屏幕。

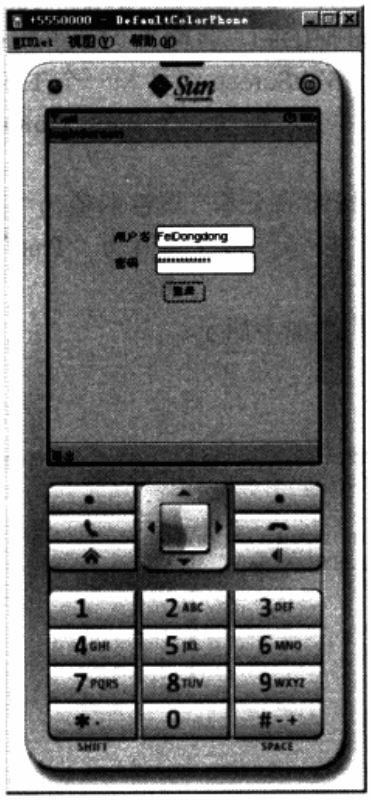


图 20-39 输入用户名和密码

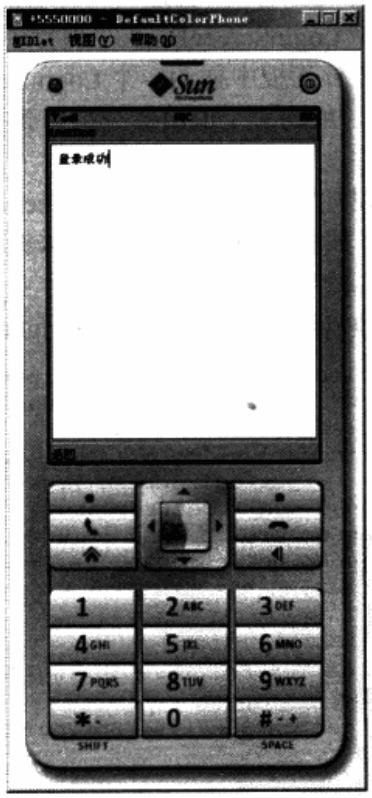


图 20-40 登录成功

和其他组件一样，登录屏幕也有自己的属性，如表 20-9 所示。

Download at http://www.pin5i.com/

表 20-9 登录屏幕的属性说明

属 性	说 明
背景色	登录屏幕的背景颜色
前景色	登录屏幕的前景颜色
全屏	登录屏幕是否全屏显示
登录按钮的文本	登录按钮上显示的文本
登录标题	登录标题
口令标签	密码文本框标签
标题	登录屏幕的标题
使用登录按钮	是否使用登录按钮
用户名标签	用户名文本框标签
口令	默认密码
用户名	默认用户名

20.9 创建 SMS 编写器（SMSComposer）

SMS 编写器是通过使用 Short Message Service（SMS）提供的一种机制和用户接口来发送短信息的定制组件。该组件利用 JSR-120 Wireless Messaging API，但此 API 并不是 MIDP 2.0 规范的一部分，所以此定制组件只能部署在内置支持 JSR-120 的设备上。

下面使用 SMS 编写器创建一个 Mobile 应用程序。

❶ 在 NetBeans 集成开发环境中新建一个“Mobile 应用程序”项目，项目名称为“SMSComposerDemo”，不创建 Hello MIDlet，仿真平台选择“Sun Java Wireless Toolkit 2.5.2 for CLDC”。在“SMSComposerDemo”项目中新建一个 Visual MIDlet，名称为“SMSComposerMidlet”。

❷ 从组件面板中拖动一个 SMSComposer 组件和一个 Alert 组件到流设计器中，向 smsComposer 组件中添加一个退出命令按钮。

❸ 在流设计器中选中 alert 组件，在属性窗口中修改其字符串属性的值为“信息已发出!”，修改其超时属性的值为“2000”。

❹ 用鼠标连接各个组件，最终流设计如图 20-41 所示。

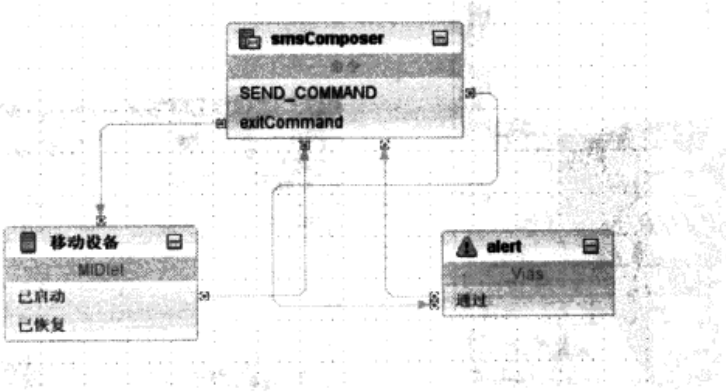


图 20-41 SMSComposerMidlet 最终流设计

❺ 为了演示发送信息的结果，需要打开 WMA 控制台。选择主菜单中的“工具”→“Java 平台”命令，在打开的“Java 平台管理器”对话框中选择“Sun Java Wireless Toolkit 2.5.2 for CLDC”平台。选择“工具和扩展”标签，单击“打开实用程序”按钮。在打开的“工具集”

Download at http://www.pin5i.com/

对话框中，选中“WMA 控制台”，单击“启动”按钮，如图 20-42 所示。

❻ 运行项目 SMSComposerDemo，在打开的模拟器中启动 SMSComposerMidlet。在“电话号码”文本框中输入“5550000”，在“信息”文本框中输入要发送的信息，此处为“预祝 2008 北京奥运会圆满成功!”，如图 20-43 所示。



图 20-42 “工具集”对话框

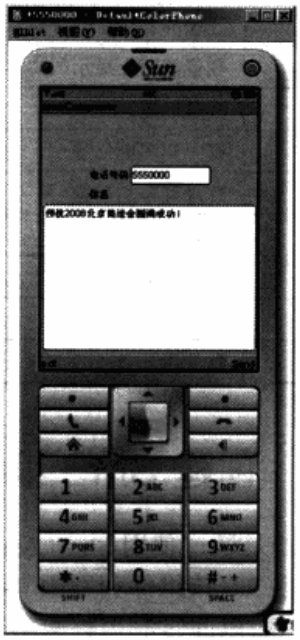


图 20-43 输入电话号码和信息

❼ 单击“发送”按钮，当信息发出后会出现提示，如图 20-44 所示。此时，在 WMA 控制台中会出现刚才发送的信息，如图 20-45 所示。



图 20-44 信息已发出

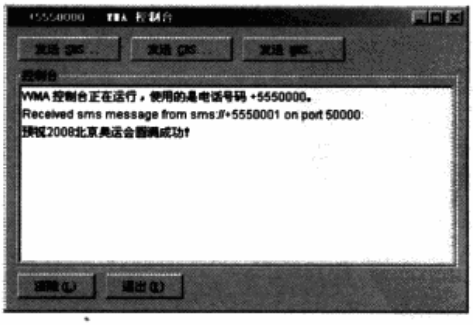


图 20-45 WMA 控制台

SMS 编写器的属性如表 20-10 所示。

Download at http://www.pin5i.com/

表 20-10 SMS 编写器的属性说明

[redacted]	[redacted]
背景色	SMS 编写器的背景颜色
前景色	SMS 编写器的前景颜色
全屏	SMS 编写器是否全屏显示
消息	发送信息文本框的标签
电话号码	电话号码文本框的标签
标题	SMS 编写器的标题
自动发送	是否自动发送信息
消息	默认信息内容
电话号码	默认电话号码
端口号	发送信息的端口号

20.10 创建列表（List）

List 是包含有选择列表的 Screen 的子类，同时实现了 Choice 接口。一个列表中可以包含若干列表元素，当列表显示在用户面前时，用户可以通过各个列表元素之间的滚动进行选择来与应用程序交互。

列表有三种类型：Choice.EXCLUSIVE、Choice.MULTIPLE 和 Choice.IMPLICIT。开发人员可以根据需要进行选择，在下面的例子中可以看出这三种类型的区别。

❶ 在 NetBeans 集成开发环境中新建一个“Mobile 应用程序”项目，项目名称为“ListDemo”，不创建 Hello MIDlet，仿真平台选择“Sun Java Wireless Toolkit 2.5.2 for CLDC”。在“ListDemo”项目中新建一个 Visual MIDlet，名称为“ListMidlet”。

❷ 从组件面板中的屏幕类别下向流设计器中拖动 4 个 List 组件。

❸ 在属性窗口中，修改 4 个 List 组件的实例名称并分别“mainList”、“excList”、“impList”和“mulList”。设置 excList 的“类型”属性为“EXCLUSIVE”并设置“mullist”的“类型”属性为“MULTIPLE”。

❹ 再分别向每个 List 组件中拖动三个 List Element 组件。再拖动三个 Back Command 命令按钮分别到 excList、impList 和 mulList 中，拖动一个退出命令按钮到 mainList 中。在属性窗口修改 mainList 中 List Element 的“字符串”属性分别为“EXCLUSIVE”、“MULTIPLE”和“IMPLICIT”。

❺ 为了增添视觉效果，可以为 List Element 添加图像。右键单击导航窗口的资源节点，在弹出的菜单中选择“新建/添加”→“图像”命令，将图像路径指定为预先准备好的图像文件所在位置。在导航窗口分别选中 EXCLUSIVE、MULTIPLE 和 IMPLICIT 节点，并在属性窗口中设置“图像”属性的值为 image。

❻ 用鼠标按如图 20-46 所示的方式连接各个组件。

❼ 按 F6 键运行主项目，在模拟器窗口中启动 ListMidlet。在出现的列表中用上下方向键选择其中一项并按选择键，可以分别切换到三个不同类别的列表，如图 20-47 所示。图 20-48、图 20-49、图 20-50 分别展示三种不同类别列表的效果。

Download at http://www.pin5i.com/

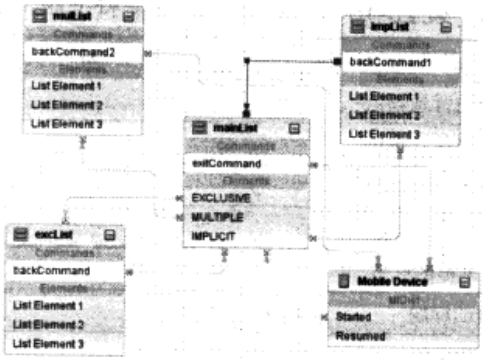


图 20-46 ListMidlet 流设计结果



图 20-47 选中并按选择键



图 20-48 EXCLUSIVE 类型效果

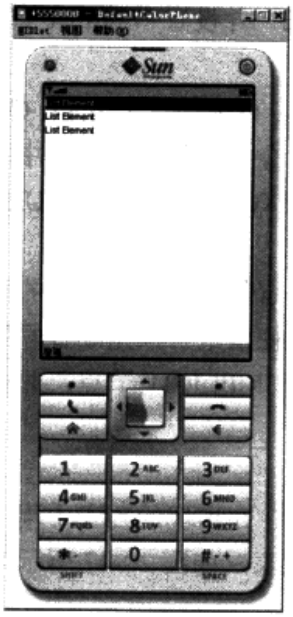


图 20-49 IMPLICIT 类型效果

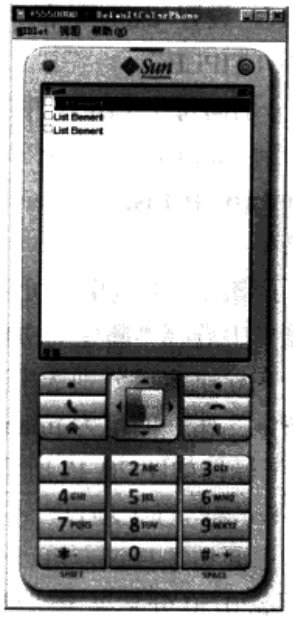


图 20-50 MULTIPLE 类型效果

Download at http://www.pin5i.com/

表 20-11 和表 20-12 分别列出了列表组件和列表元素的常用属性。

表 20-11 List 的属性说明

属 性	说 明
标题	列表的标题
类型	列表的类型，默认为 IMPLICIT
适应策略	设定当文字内容过长时应采取的策略。TEXT_WRAP_DEFAULT 表示应用程序没有优先权来决定如何处理而由具体实现使用自己默认设置。TEXT_WRAP_OFF 表示文本元素内容应该被限制在一行之内。TEXT_WRAP_ON 表示如果可利用的内容空间不足的话文本内容可以自动换行
选择命令	类型为 IMPLICIT 列表的选择操作所使用的命令按钮。IMPLICIT 类型的列表默认有一个预定义的命令按钮 List.SELECT_COMMAND。使用该属性可以将其默认命令按钮设置为其他命令按钮
基于索引的切换	如果设置为 true，生成的代码将使用 switch()构造方法来处理选择的内容

表 20-12 List Element 的属性说明

属 性	说 明
字符串	列表元素显示的文本内容
图像	文本内容前的图像
已选定	在初始状态是否为选定状态
操作	当用户按下选择键时执行的操作

20.11 创建文件浏览器

文件浏览器（FileBrowser）是一个定制组件，其提供了用于操作设备文件系统的用户接口。文件浏览器提供了浏览器设备存储内容的基本功能。FileBrowser 组件利用 FileConnection API，而此 API 是 Java ME 平台 PDA 可选包 JSR-75 规范的一部分。JSR-75 并不是包含在 MIDP 2.0 规范中，所以 FileBrowser 组件只能在支持 JSR-75 的设备上使用。

下面使用 FileBrowser 组件创建一个 Java ME 应用程序。

❶ 在 NetBeans 集成开发环境中新建一个“Mobile 应用程序”项目，项目名称为“FileBrowserDemo”，不创建 Hello MIDlet，仿真平台选择“Sun Java Wireless Toolkit 2.5.2 for CLDC”。在“FileBrowserDemo”项目中新建一个 Visual MIDlet，名称为“FileBrowserMidlet”。

❷ 从组件面板中拖动一个 FileBrowser 组件、一个 TextBox 组件和流类别下的入口点组件到流设计器中。

❸ 从组件面板拖动一个退出命令按钮到 FileBrowser 组件上，拖动一个返回命令组件到 TextBox 组件上。

❹ 选中 fileBrowser 组件，在属性窗口中修改其“过滤器”属性的值为“.txt”，其含义是只显示后缀名为 txt 的文件。

❺ 用鼠标连接流设计器中的各个组件，最终流设计如图 20-51 所示。

❻ 在 method 组件上单击鼠示右键，选择弹出菜单中的“转至源”命令，转到源代码视图。将如下代码添加到 method 方法中：

```
public void method() {
    // write pre-action user code here
    try {
        FileConnection textFile = fileBrowser.getSelectedFile();
        InputStream fis = textFile.openInputStream();
        byte[] bt = new byte[1024];
        int length = fis.read(bt,0,1024);
```

Download at http://www.pin5i.com/


```
            fis.close();
            if(length>0){
                getTextBox().setString(new String(bt,0,length));
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        }
        switchDisplayable(null, getTextBox());
        // write post-action user code here
    }
```

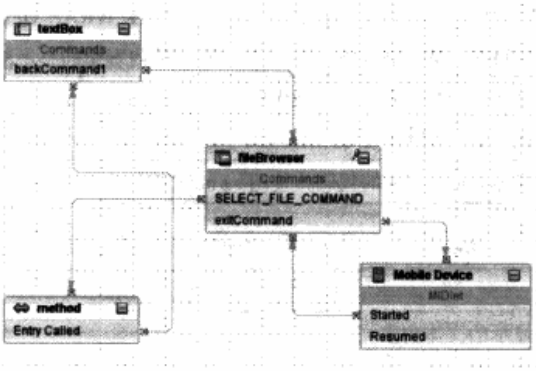


图 20-51 FileBrowserMidlet 最终流设计

❼ 为了演示文件浏览器效果，向 WTK 存储目录中添加一些文件。WTK 存储目录一般位于 X:\Documents and Settings\Username\j2mewtk\2.5.2\appdb\DefaultColorPhone\filesystem，其中 X 为系统盘，Username 为系统用户名。本例中添加一个 documents 文件夹和一个 photos 文件夹，在 documents 文件夹下添加一个名为 hello.txt 的文本文件和一个名为 java.png 的图像文件。

❽ 运行项目 FileBrowserMidlet，在打开的模拟器中启动应用程序 FileBrowserDemo。出现“是否要读取本地文件”的提示，选择“是”，出现如图 20-52 所示界面。

❾ 使用向上向下方向键选中 documents 目录，单击选择按钮。再次出现“是否要读取本地文件”的提示，选择“是”。出现如图 20-53 所示的界面。由于设置了过滤器，所以 java.png 文件没有显示出来。



图 20-52 浏览器文件夹

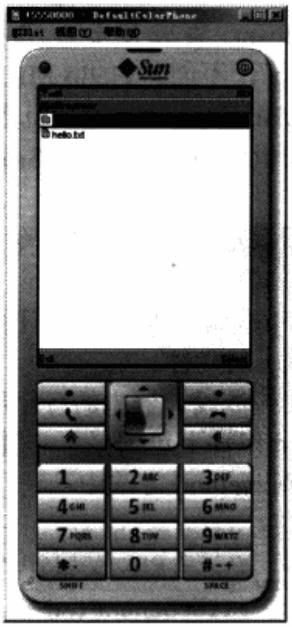


图 20-53 documents 文件夹中的文件

Download at http://www.pin5i.com/

⑩ 选中 hello.txt 文件，单击选择按钮，hello.txt 文件中的内容就会在文本框中显示出来。表 20-13 列出了文件浏览器组件的常用属性：

表 20-13 FileBrowser 的属性说明

属 性	说 明
过滤器	设置 FileBrowser 的文件过滤器
标题	FileBrowser 的标题

20.12 创建 PIM 浏览器（PIMBrowser）

PIM Browser 是一个定制组件，其提供一种简便的方式来访问标准移动设备的功能，如日历、通讯录。Java ME 应用程序可以通过 PIM Browser 组件来访问存储在本地的个人信息数据。PIMBrowser 组件利用 PIM（Personal Information Management）API，而此 API 是 Java ME 平台 PDA 可选包 JSR-75 规范的一部分。JSR-75 并不是包含在 MIDP 2.0 规范中，所以 FileBrowser 组件只能在支持 JSR-75 的设备上使用。

下面使用 FileBrowser 组件创建一个 Java ME 应用程序。

❶ 在 NetBeans 集成开发环境中新建一个“Mobile 应用程序”项目，项目名称为“PIMBrowserDemo”，不创建 Hello MIDlet，仿真平台选择“Sun Java Wireless Toolkit 2.5.2 for CLDC”。在“PIMBrowserDemo”项目中新建一个 Visual MIDlet，名称为“PIMBrowserMidlet”。

❷ 从组件面板中拖动一个 PIMBrowser 组件、一个 SplashScreen 组件和一个 Alert 组件到流设计器中，拖动一个退出命令按钮到 pimBrowser 组件上。

❸ 选中 splashScreen 组件，在属性窗口中修改其全屏属性的值为真，修改其文本属性的值为“PIMBrowser 示例”，修改其属性的值为 2000。

❹ 在属性窗口中，修改 alert 组件的字符串属性的值为“您选择了一个联系人”。

❺ 用鼠标连接流设计器中的各个组件，最终流设计如图 20-54 所示。

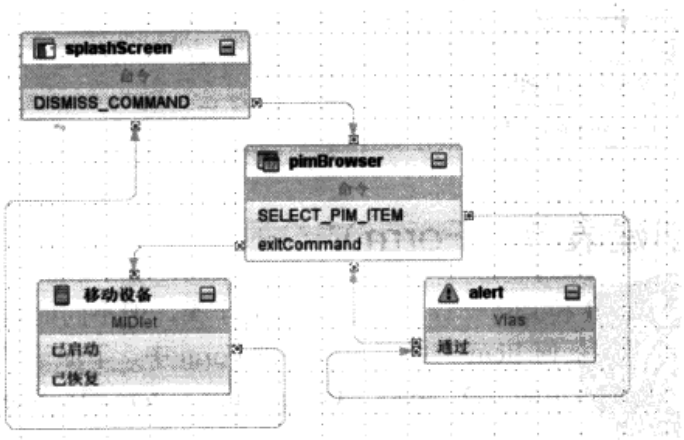


图 20-54 PIMBrowserMidlet 最终流设计

❻ 为了演示 PIMBrowser 的功能，将三个名片文件复制到 WTK 工作目录下的 appdb\DefaultColorPhone\pim\contacts\Contacts\文件夹中。本例使用了三个 VCF 文件。

❼ 运行项目 PIMBrowserDemo，在打开的模拟器中启动 PIMBrowserMidlet。在闪屏过后会出现“是否要读取用户数据”的提示，单击“是”按钮。在 PIMBrowser 屏幕中会列出包含三个联系人的列表。

Download at http://www.pin5i.com/

❽ 选中其中一个联系人，单击“菜单”按钮，如图 20-55 所示。在弹出的菜单中选择“Details”命令。在出现的屏幕中会列出选中联系人的详细信息，如图 20-56 所示。

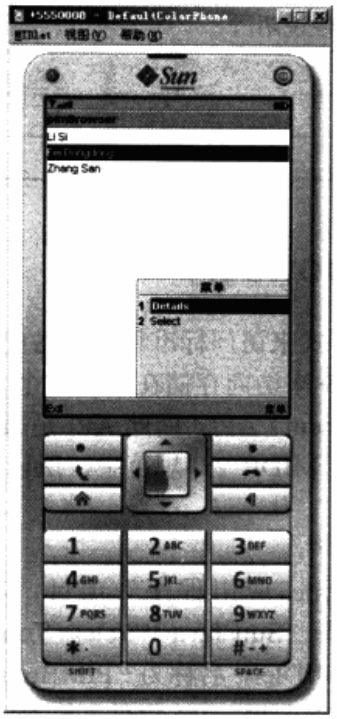


图 20-55 联系人列表

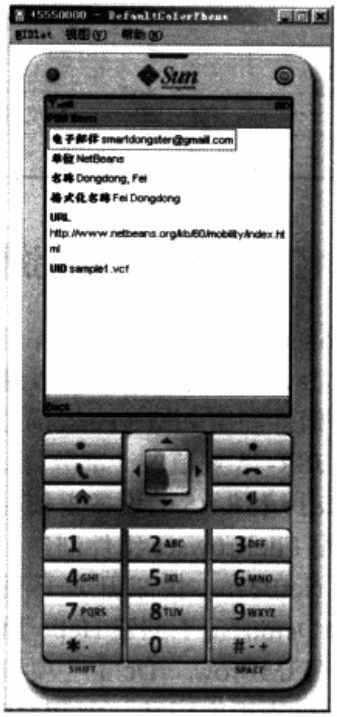


图 20-56 联系人详细信息

❾ 如果在如图 20-55 所示的菜单中，选择“Select”命令，屏幕中会出现警报信息“您选择了一个联系人”。

表 20-14 列出了 PIM 浏览器组件的常用属性：

表 20-14 PIMBrowser 的属性说明

属 性	说 明
PIM 类型	设置请求的 PIM 的类型
标题	PIMBrowser 的标题

20.13 创建表单（Form）

Form 和 TextBox、Alert 和 List 一样是 Screen 类的子类，但不同的是 Form 类是一个容器，可以容纳其他组件。可以将任何 Item 类的子类添加到 Form 中，下面这些组件的任意组合都可以被添加到 Form 中：ChoiceGroup、DateField、Gauge、ImageItem、TextField、StringItem、Spacer 和 Mobility Pack 中定义的 TableItem。

为了有效组织添加到 Form 中的组件，在 Form 中引入了布局策略。Form 中的布局是以行来组织的。通过设置 Item 的布局类型来设定其在 Form 中的对齐方式。

每个 Item 都有“最小大小”和“首选大小”两个大小概念。“最小大小”是指 Item 能够发挥作用和显示其内容的最小尺寸。“首选大小”一般是基于 Item 内容的最佳显示大小。这两个大小是由系统自动计算的，开发人员也可以自己设定首选大小。

也可以向每个 Item 组件中添加按钮，添加的按钮会在选中其被添加到的组件时显示出来。添加到 Item 组件中的命令按钮应该为“Item Command”，但添加其他类型的命令按钮也不会

Download at http://www.pin5i.com/

发生错误。Item 组件可以有一个默认命令按钮。具体设备实现通过使用平台独立的用户表示来启动 Item 组件的默认命令按钮。例如，在一个有专门选择键的设备上，按这个键可以调用组件的默认命令按钮。或者在一个触摸屏的设备上，单击 Item 组件可以调用默认命令按钮。但有些设备可能没有这样专门的按键，在这种情况下用户可以像其他命令按钮一样调用默认命令按钮。

20.13.1 使用文本字段组件（TextField）

TextField 是 Item 类的直接子类，主要用于输入文字信息。TextField 和 TextBox 有许多相同之处，TextField 只能单行显示并且在使用时应该将其添加到容器中。下面举例介绍 TextField 组件的使用。

❶ 在 NetBeans 集成开发环境中新建一个“Mobile 应用程序”项目，项目名称为“FormDemo”，不创建 Hello MIDlet，仿真平台选择“Sun Java(TM) Wireless Tookit 2.5.2 for CLDC”。在“FormDemo”项目中新建一个 Visual MIDlet，名称为“TextFieldMidlet”，放于“textfield”包中。

❷ 将一个 Form 组件拖动到流设计中，修改其“标签”属性为“请输入用户信息”。再向 form 中添加一个退出命令按钮。双击 Form 组件将其在屏幕设计器中打开。依次向 Form 的屏幕中拖入四个 TextField 组件。

❸ 在“属性”窗口中修改 textField 的标签为“用户名”，修改 textField1 的标签为“密码”并单击约束后的“定制”按钮在“约束”对话框中设置其输入限制为“任意”和“口令”，如图 20-57 所示。修改 textField2 的标签为“电话”并修改其“约束”属性为“电话号码”，修改 textField3 的标签为“电子邮件”并修改其“约束”属性为“电子邮件地址”。form 屏幕的最终效果如图 20-58 所示。

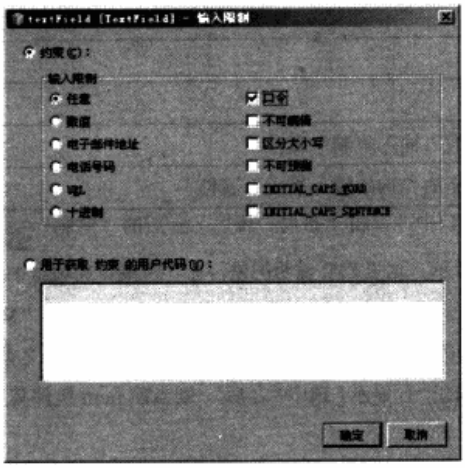


图 20-57 输入约束对话框

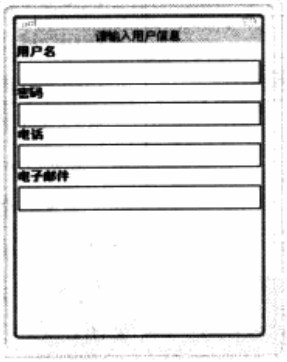


图 20-58 form 的屏幕设计结果

❹ 返回到流设计器视图，连接移动设备元素的已启动端口到 form 上，连接 form 的 exitCommand 端口到移动设备元素上。

❺ 按 F6 键运行主项目，在打开的模拟器窗口中启动 TextFieldMidlet，在各个文本字段中输入信息，如图 20-59 所示。

TextField 拥有其他 Item 组件共有的属性，也有自己特殊的属性，表 20-11 列出了 TextField 的属性。

Download at http://www.pin5i.com/



图 20-59 TextFieldMidlet 运行结果

表 20-15 TextField 的属性说明

属 性	说 明
标签	标识 TextField 的文本
文本	TextField 的文字内容
最大大小	TextField 所能容纳的最大字符数
输入限制	TextField 中文本的约束。同 TextBox 的“输入限制”属性
初始输入模式	TextField 的初始输入模式。同 TextBox 的“初始输入模式”属性
布局	TextField 的布局。可以在“属性”窗口中，单击“布局”属性名称后的“定制”，在打开的“布局”对话框中设置，如图 20-60 所示。这里需要说明的是，在“常规”选项组中选择“默认”复选框会使用具体设备的默认设置，此时所有其他选项变为不可用。如果选中“MIDP 2”复选框，当前 Item 组件会使用新的 MIDP 2.0 规则。在“插入新行”选项组选中“之前”，则当前 Item 组件会在其前面的组件之后的新的一行显示，选中“之后”，则当前 Item 组件其后的 Item 组件会在新的一行显示。 如果选中“缩小”或“展开”选项组中的“水平”或“垂直”，则 Item 组件会根据其占用空间的情况在水平或垂直方向缩小或放大以适应所占用的空间
首选高度	TextField 组件的首选高度。在“属性”窗口中单击“首选调试”属性名称后的“定制”按钮，打开“首选高度”对话框。取消“解锁”复选框并在“高度”文本框中输入数值，这样就将首选大小锁定，其数值将保持不变
首选宽度	TextField 组件的首选宽度，设置方法同首选高度
默认命令	TextField 的默认命令按钮。可以从下拉列表框中选择已添加到当前 TextField 组件的命令按钮。

Download at http://www.pin5i.com/

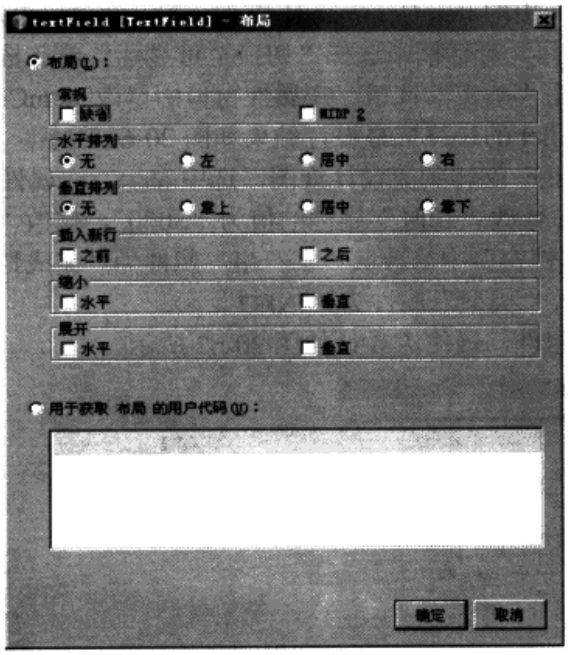


图 20-60 “布局”对话框

20.13.2 使用字符串项组件（StringItem）

StringItem 是用于显示文字信息的 Item 组件，其文字内容不能被用户编辑，但其标签和内容可以被应用程序修改。现在以实例讲解 StringItem 组件的使用。

❶ 在 NetBeans 集成开发环境中，在已创建的“FormDemo”项目中新建一个 Visual MIDlet，名称为“StringItemMidlet”，放于“stringitem”包中。

❷ 从组件面板中拖动一个 Form 组件和两个 Alert 组件到流设计器中，向 form 中添加一个退出命令按钮。

❸ 在流设计器中，双击 form 将其在屏幕设计器中打开。向 form 的设备屏幕中依次拖入四个 StringItem 组件，依次向 stringItem2 和 stringItem3 中分别拖入一个项命令按钮。

❹ 选中 stringItem，在属性窗口中修改其“标签”属性为“这里是标签”并修改其“文本”属性为“这里是 StringItem 组件的文本内容”。在“字体”下拉列表框中选择“新的字体资源”，在“导航”窗口选中“资源”节点下的 font，在属性窗口中其修改“类型”属性的值为“定制”，修改“字样”属性的值为“系统”，修改“样式”属性的值为“斜体”和“下画线”。

❺ 选中 stringItem1，在属性窗口中修改其“标签”属性为“普通”并修改其“文本”属性为“StringItem 的外观为 PLAIN”。

❻ 选中 stringItem2，在属性窗口中修改其“标签”属性为“链接”并修改其“文本”属性为“StringItem 的外观为 HYPERLINK”。在“外观”下拉列表框中选择“HYPERLINK”。在“默认命令”下拉列表框中选择“itemCommand”。

❼ 选中 stringItem3，在属性窗口中修改其“标签”属性为“按钮”并修改其“文本”属性为“StringItem 的外观为 BUTTON”。在“外观”下拉列表框中选择“BUTTON”。在“默认命令”下拉列表框中选择“itemCommand1”。

❽ 在屏幕设计器中选中的 itemCommand 组件，在属性窗口单击“操作”属性后面的定制按钮，在打开的“操作”对话框中，选择“切换至可显示”单选按钮并在“目标屏幕”下拉列表框中选择“alert”，单击“确定”按钮。在属性窗口中修改 itemCommand“标签”属性为“转到”。

Download at http://www.pin5i.com/

❾ 在屏幕设计器中选中的 itemCommand1 组件，在属性窗口单击“操作”属性后面的“定制”按钮，在打开的“操作”对话框中选择“切换至可显示”单选按钮并在“目标屏幕”下拉列表框中选择 alert1，单击“确定”按钮。在属性窗口中修改 itemCommand1 的“标签”属性为“点击”。StringItemMidlet 的最终屏幕设计结果如图 20-61 所示。

❿ 按 Alt + Shift + Right 键转到流设计视图，选中 alert，在属性窗口中修改其“字符串”属性为“转到指定链接”，修改“超时”属性的值为“3000”，修改“类型”属性为“INFO”。

⓫ 选中 alert1，在属性窗口中修改其“字符串”属性为“完成特定的动作”，修改“超时”属性的值为“3000”，修改“类型”属性为“INFO”。

⓬ 用鼠标连接各个组件，最终流设计如图 20-62 所示。

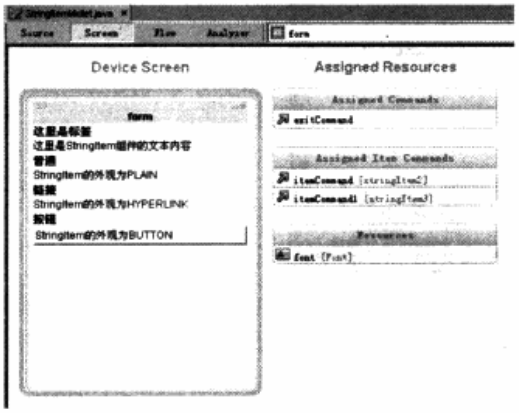


图 20-61 StirngItemMidlet 屏幕设计结果

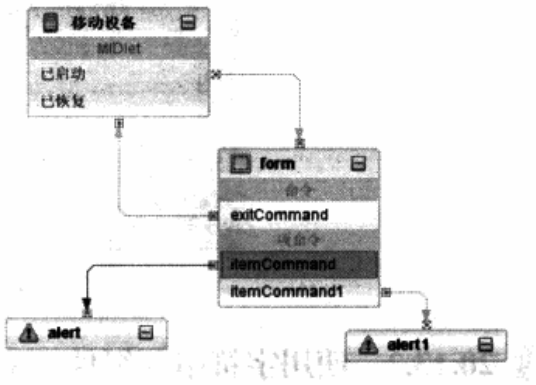


图 20-62 StringItemMidlet 最终流设计

⓭ 运行 FormDemo，在打开的模拟器窗口中启动 StringItemMidlet。可以看到使用不同外观和使用特定字体的效果。使用上下方向导航到“链接”字符串项，按下选择键后的结果如图 20-63 所示，若按“转到”键可以切换到警报屏幕。



图 20-63 StringItemMidlet 运行结果

Download at http://www.pin5i.com/

下面对 StringItem 组件的属性进行简要介绍，其属性如表 20-16 所示。

表 20-16 StringItem 的属性说明

属性	说明
标签	StringItem 组件的标签
文本	StringItem 组件的文本内容
外观	StringItem 组件的外观模式，有 PLAIN、HYPERLINK 和 BUTTON 三种
字体	StringItem 组件中文本的字体格式
布局	StringItem 组件的布局方式，同 TextField 组件的“布局”属性
首选高度	StringItem 组件的首选高度，同 TextField 组件的“首选高度”属性
首选宽度	StringItem 组件的首选宽度，同 TextField 组件的“首选宽度”属性
默认命令	StringItem 组件的默认命令，同 TextField 组件的“默认命令”属性

20.13.3 使用图像项组件（ImageItem）

ImageItem 组件用于显示图像，图像可以为可变的或不可变的。下面以实例讲解 ImageItem 组件的使用。

❶ 首先准备三张图片，此例中图像的文件名分别为 leaf.png（用于显示不变图像）、image1.png 和 image2.png（用于显示可变图像）。

❷ 在 NetBeans 集成开发环境中，在已创建的“FormDemo”项目中新建一个 Visual MIDlet，名称为“ImageItemMidlet”，放于“imageitem”包中。将三张图片文件复制到 src/imageitem 文件夹中。

❸ 拖动一个 Form 组件到流设计器中，向 form 中添加一个退出命令按钮和两个 ImageItem 组件。

❹ 在属性窗口中，修改 imageItem 的标签属性的值为“不变图像”，修改 imageItem1 的标签属性的值为“可变图像”。

❺ 在导航窗口中，新建一个 Image 资源，并设置其资源路径属性为 leaf.png 的文件路径。设置 imageItem 的“图像”属性为 image，再设置其“布局”属性，在“布局”对话框中选择“水平排列”选项组为“居中”，选择“插入新行”选项组为“之后”。同理，修改 imageItem1 布局的水平排列为“居中”。

❻ 切换到源视图，在源代码编辑器中修改 ImageItemMidlet.java，修改后的代码如下所示（未修改的方法代码和注释已省略）：

```
package imageitem;

import java.io.IOException;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class ImageItemMidlet extends MIDlet implements CommandListener {

    private boolean midletPaused = false;

    private Form form;
    private ImageItem imageItem1;
    private ImageItem imageItem;
    private Command exitCommand;
    private Image image;
    private Image[] images;//定义 images 数组
```

Download at http://www.pin5i.com/


```
    public ImageItemMidlet() {…}
    private void initialize() {…}
    public void startMIDlet() {…}
    public void resumeMIDlet() {…}
    public void switchDisplayable(Alert alert, Displayable nextDisplayable) {…}
    public void commandAction(Command command, Displayable displayable) {…}
    public Form getForm() {… }
    public ImageItem getImageItem() {…}
    public ImageItem getImageItem1() {
        if (imageItem1 == null) {
            // write pre-init user code here
            imageItem1 = new ImageItem("\u53EF\u53D8\u56FE\u50CF", null,
                         ImageItem.LAYOUT_CENTER, "<Missing Image>");
            // write post-init user code here
            new Thread(new Runnable(){//启动新线程，交替显示两个图像
                public void run() {
                    int index = 0;
                    while (true){
                        getImageItem1().setImage(get_images()[index++]);
                        if(index==2){
                            index = 0;
                        }
                        try {
                            Thread.sleep(300);
                        } catch (InterruptedException ex) {
                            ex.printStackTrace();
                        }
                    }
                }
            }).start();
        }
        return imageItem1;
    }
    //获得 images 图像数组
    public Image[] get_images(){
        if(images==null){
            images = new Image[2];
            try {
                images[0] = Image.createImage("/imageitem/image1.png");
                images[1] = Image.createImage("/imageitem/image2.png");
            } catch (IOException ex) {
                ex.printStackTrace();
            }
        }
        return images;
    }
    public Command getExitCommand() {…}
    public Image getImage() {…}
    public Display getDisplay () {…}
    public void exitMIDlet() {…}
    public void startApp() {…}
    public void pauseApp() {…}
    public void destroyApp(boolean unconditional) {
    }
}
```

❼ 切换到流设计视图，用鼠标连接移动设备元素的已启动端口到 form 上，连接 form 上的 exitComand 端口到移动设备元素上。

❽ 按 F6 键运行主项目，在打开的模拟器窗口中启动 ImageItemMidlet。可以看到一个不变的图像和一个会变动的图像，如图 20-64 所示。

Download at http://www.pin5i.com/



图 20-64 ImageItemMidlet 运行结果

ImageItem 组件的属性如表 20-17 所示。

表 20-17 ImageItem 的属性说明

属 性	说 明
标签	ImageItem 组件的标签
图像	ImageItem 组件使用的图像资源
替换文本	如果由于某种原因图像不能正常显示时将显示替换文本
外观	ImageItem 组件的外观模式，有 PLAIN、HYPERLINK 和 BUTTON 三种
布局	ImageItem 组件的布局方式，同 TextField 组件的“布局”属性
首选高度	ImagegItem 组件的首选高度，同 TextField 组件的“首选高度”属性
首选宽度	ImageItem 组件的首选宽度，同 TextField 组件的“首选宽度”属性
默认命令	ImageItem 组件的默认命令，同 TextField 组件的“默认命令”属性

Image 组件的属性如表 20-18 所示。

表 20-18 Image 的属性说明

属 性	说 明
资源路径	图像文件的路径

20.13.4 使用标尺组件（Gauge）

Gauge 通常用来表示任务的进展情况或者在给定范围内设定某种参数和数值。Gauge 组件有两种模式：交互模式和非交互模式。交互模式的 Gauge 用户可以改变其值；非交互模式的 Gauge 只能用来显示而不能被更改。下面以实例讲解 Gauge 组件的使用。

❶ 在 NetBeans 集成开发环境中，在已创建的“FormDemo”项目中新建一个 Visual MIDlet，

Download at http://www.pin5i.com/

名称为“GaugeMidlet”，放于“gauge”包中。

❷ 向流设计器拖入一个 Form 组件，再 form 中添加一个退出命令按钮和两个 Gauge 组件。

❸ 在导航窗口中，选中 gauge 节点，在属性窗口中修改其“标签”属性为“非交互模式”，“最大值”属性为“5”，“值”属性为“0”。用同样的方法修改 gauge1 的“标签”属性为“交互模式”，选中“交互”复选框，修改“最大值”属性为“5”，值属性为“0”。

❹ 切换到源视图，在源代码编辑器中修改 GaugeMidlet.java，修改后的代码如下所示（未修改的方法代码和注释已省略）：

```
package gauge;

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class GaugeMidlet extends MIDlet {

    private boolean midletPaused = false;

    private Form form;
    private Gauge gauge;
    private Gauge gauge1;
    private boolean isRun;
    private int maxValue;
    private int count = 1;

    public GaugeMidlet() {
    }
    private void initialize() {…}
    public void startMIDlet() {…}
    public void resumeMIDlet() {…}
    public void switchDisplayable(Alert alert, Displayable nextDisplayable) {…}
    public Form getForm() {…}
    public Gauge getGauge1() {…}
    public Gauge getGauge() {
        if (gauge == null) {
            // write pre-init user code here
            gauge = new Gauge("\u4EA4\u4E92\u6A21\u5F0F", true, 5, 0);
            // write post-init user code here
             start();
            this.maxValue = gauge.getMaxValue();
            new Thread(new Runnable(){
                public void run() {
                    while (isRun){
                        int theValue = gauge.getValue() + count;
                        if(theValue==maxValue ||theValue==0 ){
                            count = -count;
                        }
                        gauge1.setValue(theValue);
                        try {
                            Thread.sleep(500);
                        } catch (InterruptedException ex) {
                            stop();
                        }
                    }
                }

            }).start();
        }
        return gauge;
    }
    //</editor-fold>
    public void start(){
        isRun = true;
    }

    public void stop(){
```

Download at http://www.pin5i.com/


```
            isRun = false;
        }
        public Display getDisplay () {…}
        public void exitMIDlet() {…}
        public void startApp() {…}
        public void pauseApp() {…}
        public void destroyApp(boolean unconditional) {
        }
    }
```

❺ 切换到流设计视图，用鼠标连接移动设备元素的已启动端口到 form 上，连接 form 的 exitCommand 端口到移动设备元素上。

❻ 按 F6 键运行主项目，在打开的模拟器窗口中启动 GaugeMidlet。两个不同模式的 Gauge 运行结果如图 20-65 所示。

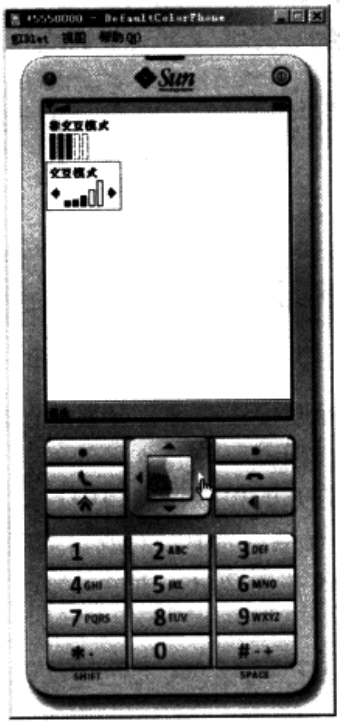


图 20-65 GaugeMidlet 运行结果

Gauge 组件的属性如表 20-19 所示。

表 20-19 Gauge 的属性说明

属 性	说 明
标签	Gauge 组件的标签
交互	Gauge 组件的交互模式，设置为 true 表示为交互模式，否则为非交互模式
最大值	Gauge 组件的最大值的大小
值	Gauge 组件的当前值的大小
布局	Gauge 组件的布局方式，同 TextField 组件的“布局”属性
首选高度	Guage 组件的首选高度，同 TextField 组件的“首选高度”属性
首选宽度	Guage 组件的首选宽度，同 TextField 组件的“首选宽度”属性
默认命令	Gauge 组件的默认命令，同 TextField 组件的“默认命令”属性

Download at http://www.pin5i.com/

20.13.5 使用日期字段组件（DateField）

DateField 组件一种是用来显示时间的可编辑组件，也就是说用户可以修改其值。下面以实例讲解 Gauge 组件的使用。

❶ 在 NetBeans 集成开发环境中，在已创建的“FormDemo”项目中新建一个 Visual MIDlet，名称为“DateFieldMidlet”，放于 datefield 包中。

❷ 向流设计器拖入一个 Form 组件，再 form 中添加一个退出命令按钮和一个 DateField 组件。

❸ 因为 DateField 组件在构造时默认使用当前日期，可以在属性窗口中修改其“日期”属性，在打开的“日期”对话框中直接输入日期的值，或者使用代码获取日期，如图 20-66 所示。

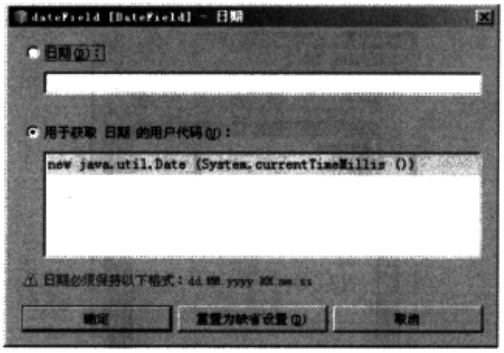


图 20-66 “日期”对话框

❹ 切换到流设计视图，用鼠标连接移动设备元素的已启动端口到 form 上，连接 form 上的 exitCommand 端口到移动设备元素上。

❺ 按 F6 键运行主项目，在打开的模拟器窗口中启动 DateFieldMidlet，如图 20-67。在 DateField 组件中会分别显示时间和日期，选中时间或日期然后按下选择键，可以在转到的屏幕中修改时间和日期，分别如图 20-68 和图 20-69 所示。修改完成后按“保存”键即可设置新的时间或者日期。



图 20-67 启动 DateFieldMidle

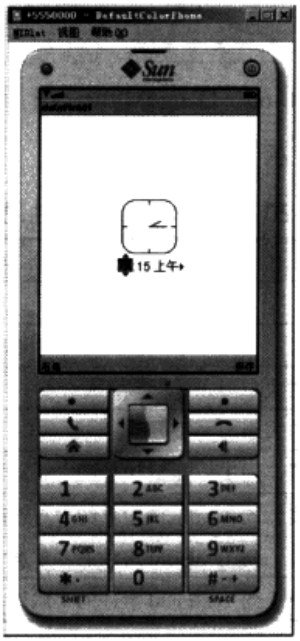


图 20-68 修改时间

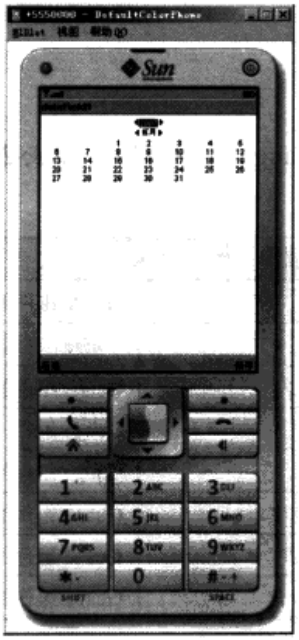


图 20-69 修改日期

Download at http://www.pin5i.com/

DateField 组件的属性如表 20-20 所示。

表 20-20 DateField 的属性说明

日期	DateField 组件默认的日期值
标签	DateField 组件的标签
输入模式	Gauge 组件的输入模式，有 DATE_TIME、DATE 和 TIME 三种。DATE_TIME 表示在 DateField 中显示日期和时间；DATE 表示仅显示日期；TIME 表示仅显示时间
布局	DateField 组件的布局方式，同 TextField 组件的“布局”属性
首选高度	DateField 组件的首选高度，同 TextField 组件的“首选高度”属性
首选宽度	DateField 组件的首选宽度，同 TextField 组件的“首选宽度”属性
时区	DateField 表示时间所在的时区，可以在时区对话框中进行选择或者输入代码获取
默认命令	DateField 组件的默认命令，同 TextField 组件的“默认命令”属性

20.13.6 使用选项组组件（ChoiceGroup）

ChoiceGroup 组件是一组包含可选择元素的窗体项组件，其功能类似和 List 类似，ChoiceGroup 类和 List 类都实现了 Choice 接口。但两者也有区别，List 可以作为独立的屏幕显示而 ChoiceGroup 只能被添加到其他容器中。此外，ChoiceGroup 的类型可以为 Choice. EXCLUSIVE、Choice. MULTIPLE 和 Choice. POPUP，而 List 的类型可以为 Choice. EXCLUSIVE、Choice. MULTIPLE 和 Choice.IMPLICIT。下面以实例讲解 ChoiceGroup 组件的使用。

❶ 在 NetBeans 集成开发环境中，在已创建的“FormDemo”项目中新建一个 Visual MIDlet，名称为“ChoiceGroupMidlet”，放于 choicegroup 包中。

❷ 从组件面板中拖动一个 Form 组件到流设计器中，拖动一个退出命令按钮组件到 form 中，双击 fom 将其在屏幕设计器中打开。

❸ 从组件面板中依次向 form 的屏幕中拖入三个 ChoiceGroup 组件，分别向每个 ChoiceGroup 组件中拖入四个 Choice Element 元素。在属性窗口中，修改 choiceGroup 的标签为“EXCLUSIVE”，并修改其类型为“EXCLUSIVE”。修改 choiceGroup1 的标签为“MULTIPLE”，并修改其类型为“MULTIPLE”。修改 choiceGroup2 的标签为“POPUP”，并修改其类型为“POPUP”。

❹ 将预先准备好的图像文件复制到 src 文件夹中。新建一个 Image 资源，将 image 的资源路径设置为刚才复制的图像文件路径。在导航窗口中选中一个 Choice Element 节点，在属性窗口中将其图像属性设置为 image。同样将所有 Choice Element 的图像设置为 image，此时 form 的屏幕结果如图 20-70 所示。

❺ 切换到流设计视图，连接移动设备元素的已启动端口到 form 号，连接 form 上的 exitCommand 端口到移动设备元素上。

❻ 按 F6 键运行主项目，在打开的模拟器窗口中启动 ChoiceGroupMidlet。使用上下方向键将焦点移动到 POPUP 选项组，按下选择键会弹出选择菜单，如图 20-71 所示。

ChoiceGroup 组件的属性如表 20-21 所示。

表 20-21 ChoiceGroup 的属性说明

标签	ChoiceGroup 组件的标签
类型	ChoiceGroup 组件的类型，有 EXCLUSIVE、MULTIPLE 和 POPUP 三种

Download at http://www.pin5i.com/

续表

属　性	说　明
适应策略	同 List 组件的“适应策略”属性
布局	ChoiceGroup 组件的布局方式，同 TextField 组件的“布局”属性
首选高度	ChoiceGroup 组件的首选高度，同 TextField 组件的“首选高度”属性
首选宽度	ChoiceGroup 组件的首选宽度，同 TextField 组件的“首选宽度”属性
默认命令	ChoiceGroup 组件的默认命令，同 TextField 组件的“默认命令”属性

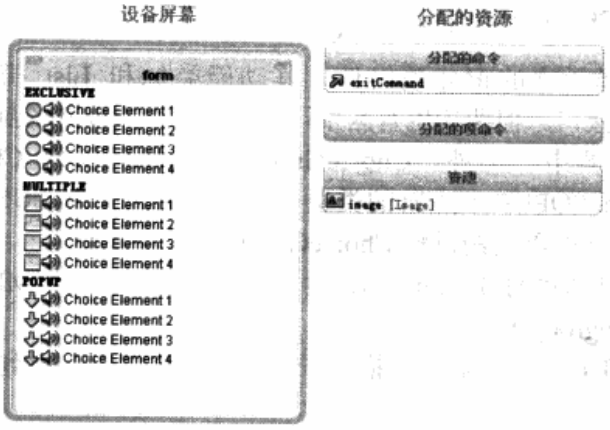


图 20-70　ChoiceGroupMidlet 屏幕设计结果

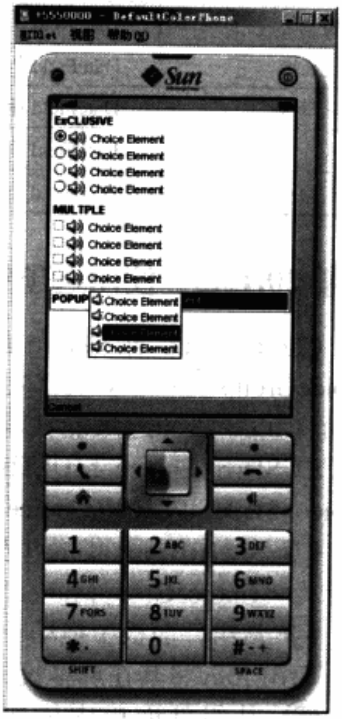


图 20-71　ChoiceGroupMidlet 运行结果

Choice Element 元素的属性如表 20-22 所示。

表 20-22　Choice Element 的属性说明

属　性	说　明
字符串	ChoiceElement 显示的文本内容
图像	文本内容前的图像
字体	ChoiceElement 显示的文本的字体
已选定	在初始状态是否处于选定状态



20.13.7　使用 TableItem 组件

可以 TableItem 组件快速创建包含多行和多列的表格，并且表格可以包含表头。表格可能在宽度和高度上都比屏幕大，用户可以使用光标在表格各个方向滚动。TableItem 通过 TableModel（表模型）模型化出表格。TableModel 是一个接口，SimapleTableModel 实现了这个接口，所以可以在 TableItem 中使用 SimapleTableModel 创建表格。下面以实例讲解 ChoiceGroup 组件的使用。

❶ 在 NetBeans 集成开发环境中，在已创建的“FormDemo”项目中新建一个 Visual MIDlet，

Download at http://www.pin5i.com/

名称为“TableItemMidlet”，放于 tableitem 包中。

❷ 从组件面板拖动一个 Form 组件和一个 TextBox 组件到流设计器中。

❸ 双击 form 将其在屏幕设计器中打开。用鼠标将 TableItem 组件从组件面板中拖动到 form 的屏幕中。选中 tableItem，在属性窗口中修改其标签为“TableItem Example”。

❹ 从组件面板中拖动一个 SimpleTableModel 到 tableItem 中。双击 simpleTableModel 可以在打开的“值”对话框中编辑其参数，如图 20-72 所示。可以单击右侧的“添加行”按钮增加一行，单击“删除行”按钮减少一行。也可以使用同样的方法修改列数。选中“使用标题”复选框表格将会包含表头。双击下面表格的单元格可以修改其内容。此处修改列数为“5”行数为“10”并选中“使用表头”复选框，依次修改表头内容为“第一列”、“第二列”、“第三列”、“第四列”和“第五列”。最后单击“确定”按钮。

❺ 切换到流设计视图，向 form 中添加一个退出命令按钮和一个项命令按钮，向 textBox 中添加一个确定按钮命令按钮。在属性窗口修改 itemCommand 的标签为“编辑”，修改 okCommand 的标签为“保存”。

❻ 如图 20-73 所示连接各个组件。

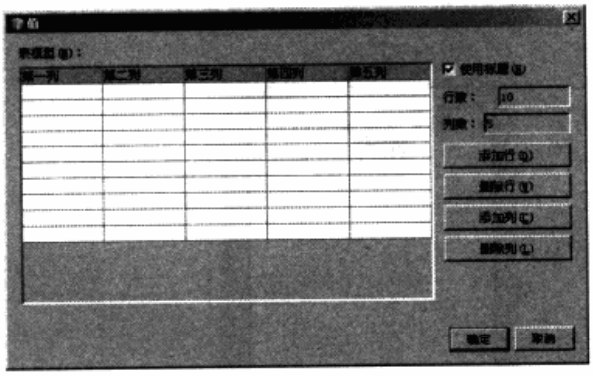


图 20-72 “值”对话框

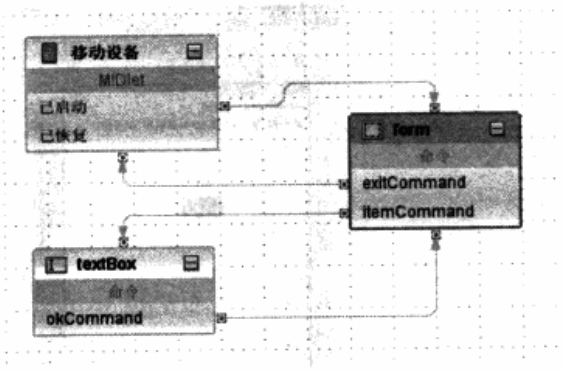


图 20-73 TableItemMidlet 流设计结果

❼ 切换到源视图，在源代码编辑器中修改 TableItemMidlet.java。修改后的 commandAction() 方法如下所示：

```
public void commandAction(Command command, Displayable displayable) {
        // write pre-action user code here
        if (displayable == form) {
            if (command == exitCommand) {
                // write pre-action user code here
                exitMIDlet();
                // write post-action user code here
            } else if (command == itemCommand) {
                // write pre-action user code here
                switchDisplayable(null, getTextBox());
                // write post-action user code here
                if (textBox != null) {
                    // 获得表格中光标的位置
                    int row = tableItem.getSelectedCellRow();
                    int column = tableItem.getSelectedCellColumn();
                 // 获得表模型中值
                Object value = tableModel.getValue(column,row);
                // 设置 textBox 字符串值
                if (value != null)
                    textBox.setString(value.toString());
                else
                    textBox.setString("");
                }
            }
        } else if (displayable == textBox) {
```

Download at http://www.pin5i.com/


```
            if (command == okCommand) {
                // write pre-action user code here
                int row = tableItem.getSelectedCellRow();
                int column = tableItem.getSelectedCellColumn();
                // 为表模型设置新值
                tableModel.setValue(column,row,textBox.getString());
                // 在设备屏幕重绘置表模型
                tableModel.fireTableModelChanged();
                switchDisplayable(null, getForm());
                // write post-action user code here
            }
        }
        // write post-action user code here
    }
```

❽ 按 F6 键运行主项目，在打开的模拟器窗口中启动 TableItemMidlet。在模拟器的屏幕中可以看到 5 列 10 行的表格，如图 20-74 所示。使用上下左右方向键可以移动光标到各个单元格上。按上“编辑”按钮然后在切换到的文本框中输入单元格的文本内容，按“保存”按钮返回到表单屏幕，可以看到表格的内容被修改了，如图 20-75 所示。

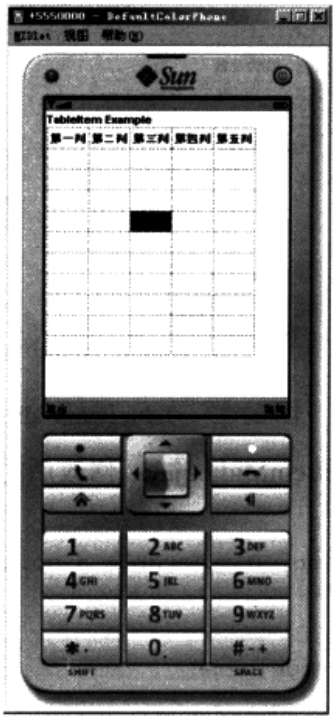


图 20-74 TableItemMidlet 运行结果

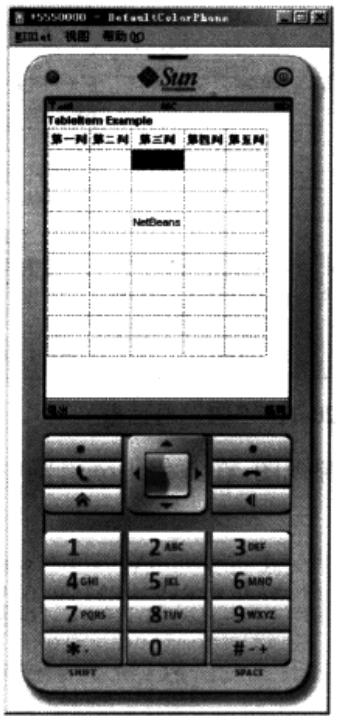


图 20-75 编辑后的表格

TableItem 组件的属性如表 20-23 所示。

表 20-23 TableItem 的属性说明

属 性	说 明
标签	TableItem 组件的标签
标题	TableItem 组件的标题，标题会显示表格的上方
表模型	TableItem 组件使用的表模型
显示边框	在表格中是否显示表格边框
布局	TableItem 组件的布局
标题字体	标题的字体

Download at http://www.pin5i.com/

续表

属　性	说　明
表头字体	表头的字体
值字体	单元格值的字体
首选高度	TableItem 组件的首选高度，同 TextField 组件的“首选高度”属性
首选宽度	TableItem 组件的首选宽度，同 TextField 组件的“首选宽度”属性
默认命令	TableItem 组件的默认命令，同 TextField 组件的“默认命令”属性

SimpleTableModel 组件的属性如表 20-24 所示。

表 20-24　SimpleTableModel 的属性说明

属　性	说　明
值	表模型的参数。其值通过“值”对话框中。其值的设置也可以通过步骤❹完成

20.13.8　使用分隔符组件（Spacer）

Spacer 是用来布局其他组件的组件。Spacer 是空白的并且是非交互的，应用程序可以设置其最小大小。下面以实例讲解 Spacer 组件的使用。

❶ 在 NetBeans 集成开发环境中，在已创建的“FormDemo”项目中新建一个 Visual MIDlet，名称为“SpacerMidlet”，放于“spacer”包中。

❷ 从组件面板拖动一个 Form 组件到流设计器中，拖动一个退出命令按钮组件到 form 中。

❸ 双击 form 将其在屏幕设计器中打开。依次向 form 的屏幕中拖动三个 StringItem 组件、一个 Spacer 组件、一个 StringItem 组件、一个 Spacer 组件和一个 StringItem 组件。双击 stringItem2，修改其内容为“添加 Spacer 后的效果”。此时 SpacerMidlet 的屏幕设计器如图 20-76 所示。

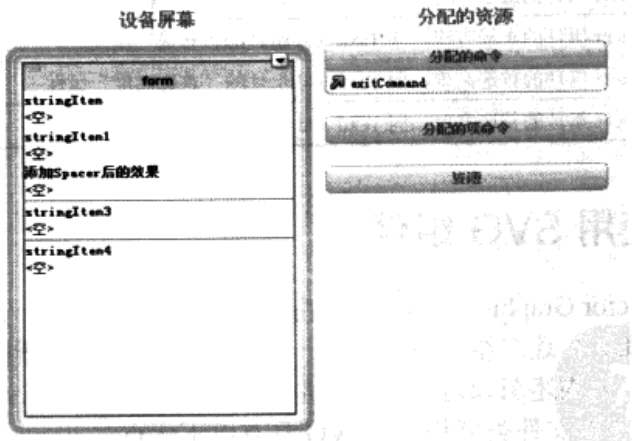


图 20-76　SpacerMidlet 屏幕设计结果

❹ 切换到流设计视图，连接移动设备元素的已启动端口到 form 上，连接 form 的 exitCommand 端口到移动设备元素上。

❺ 按 F6 键运行主项目，在打开的模拟器窗口中启动 TableItemMidlet。可以看到添加 Spacer 后两个 StringItem 组件的间隔明显变大了，如图 20-77 所示。

Download at http://www.pin5i.com/

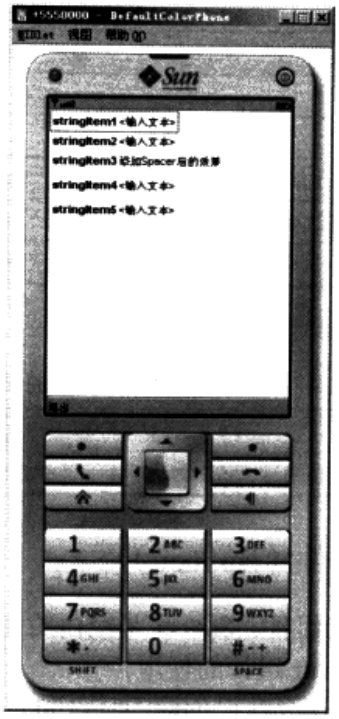


图 20-77 SpacerMidlet 运行结果

Spacer 组件的属性如表 20-25 所示。

表 20-25 Spacer 的属性说明

属 性	说 明
最小高度	Spacer 组件的最小高度
最小宽度	Spacer 组件的最小宽度，值必须为非负，单位为像素
布局	Spacer 组件的布局方式，同 TextField 组件的“布局”属性
首选高度	Spacer 组件的首选高度，同 TextField 组件的“首选高度”属性
首选宽度	Spacer 组件的首选宽度，同 TextField 组件的“首选宽度”属性

20.14 使用 SVG 组件

SVG（Scalable Vector Graphics），是“可缩放矢量图形”的意思。它是由 W3C 制定的基于可扩展标记语言（XML）描述二维矢量图型的一个开放标准。SVG 严格遵从 XML 语法，并用文本格式的描述性语言来描述图像内容，因此是一种和图像分辨率无关的矢量图形格式。

与 PNG 以及其他图像文件类型相比，SVG 具有以下优点。

- 可缩放性。SVG 能够在不同的像素分辨率、颜色深度、CPU 功率和内存下以最佳质量呈现图像。它还可以在不降低图像质量的情况下对其进行缩放。
- 大小。SVG 和 SVGZ（压缩的 SVG）文件可以比具有相似质量的 PNG 文件小几倍，从而使开发者能够在硬件资源有限的平台上创建出图形丰富的应用程序。
- 动画。SVG 可以生成脚本和动画，从而允许用户与可视化内容进行交互。
- 可搜索性。由于 SVG 采用的是 XML 代码，因此用户可以搜索在图像文件中嵌入的文本。例如，您可以在地图图像上查找街道或城镇的名称。

SVG Tiny 又称 SVGT（Scalable Vector Graphics, Tiny Profile），可以翻译为“可缩放的矢量

Download at http://www.pin5i.com/

图型标准的微型简化版本”。它是矢量图型标准——SVG 的一个子集，而主要目标是为手机等低端移动设备提供矢量图形显示格式。移动设备对 SVG 的支持正在不断地加强，但是根据移动设备的不同仍然会有所差异；而且与 PNG 相比，SVG 尚未得到广泛支持。NetBeans 6.0 支持 SVG Tiny 1.1。

20.14.1 预览 SVG 文件

在 NetBeans 集成开发环境中新建一个“Mobile 应用程序”项目，项目名称为“SVGDemo”。不创建 Hello MIDlet，默认仿真平台选择“Sun Java(TM) Wireless Tookit 2.5.2 for CLDC”。

将要使用的 SVG 文件复制到 src 文件夹中。若要预览 SVG 文件，在“项目”窗口中双击 SVG 文件，文件将会在如图 20-78 所示的窗口中打开。

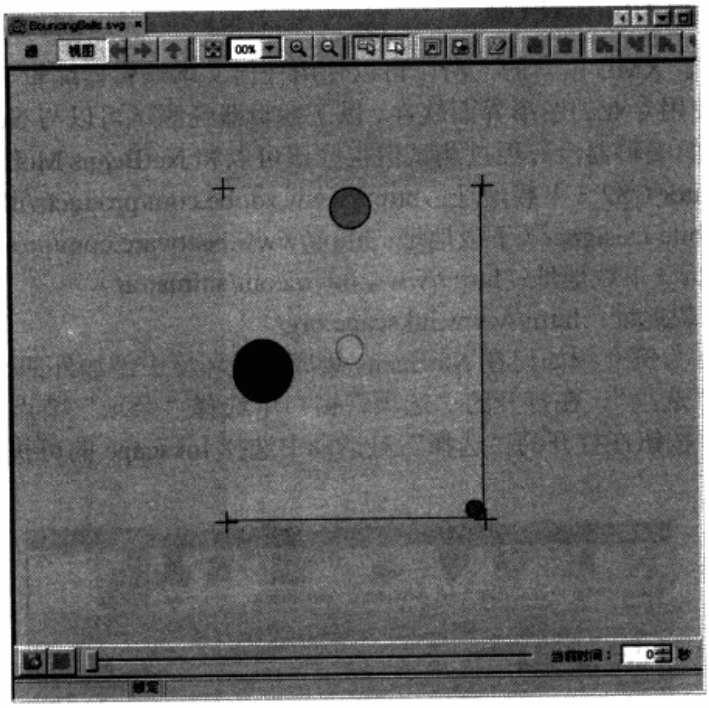


图 20-78 预览 SVG 文件

可以拖动滑杆上的滑块来预览动画各个时间段上的画面。同时可以在“当前时间”微调框中输入时间值或通过微调按钮来调整当前显示的时间（单位为秒）。在“结束时间”微调框中输入时间值或调节微调来设置动画结束时间（单位为秒）。

单击“切换显示工具提示”按钮 可以在显示工具提示和不显示工具提示之间进行切换，显示工具提示是指当鼠标停留在不同图像元素上时显示诸如类型等的提示。单击“切换突出显示”按钮 可以在突出显示和不突出显示之间切换，突出显示是指当鼠标停留在不同的元素上时，元素的整个轮廓会突出显示。单击“切换缩放”按钮 可以在图像以原始大小显示和图像以适应窗口大小显示之间切换。打开的 SVG 文件在默认情况下是锁定的，单击“解锁”按钮 ，解除对文件的锁定，这样就可以编辑 SVG 文件了。单击“切换运行动画”按钮 可以播放动画和停止播放动画。

在导航窗口中，列出了 SVG 文件的文档结构，如图 20-79 所示。同时，在组件面板中给出了常用的基本形状，可以用鼠标拖动到 SVG 文件编辑器，实现 SVG 文件的可视化编辑，如图 20-80 所示。

Download at http://www.pin5i.com/

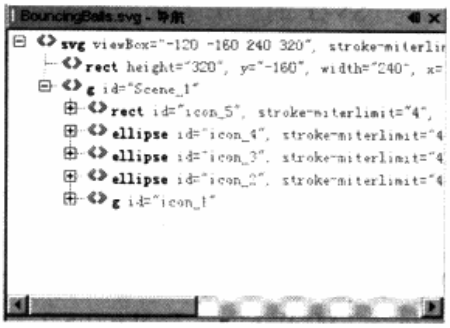


图 20-79 导航窗口

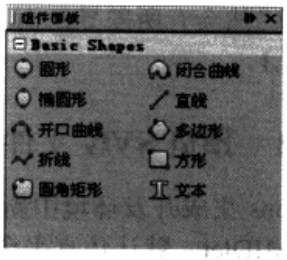


图 20-80 组件面板

20.14.2 设置外部编辑器

因为 SVG 是基于 XML 语言的，所以可以使用任何文本编辑器编辑 SVG 文件的内容。但是我们可能更愿意使用专业的图形界面软件。以下编辑器经测试可以与 NetBeans Mobility Pack 一起使用（虽然其他的编辑器没有经过测试但也应该可以和 NetBeans Mobility Pack 一起使用）：

- Adobe Illustrator CS2（下载地址：http://www.adobe.com/products/illustrator/main.html）
- Beatware Mobile Designer（下载地址：http://www.beatware.com/products/md_golive.html）
- Ikivo Animator（下载地址：http://www.ikivo.com/animator/）
- Inkscape（下载地址：http://www.inkscape.org/）

下面以 Inkscape 为例介绍如何在 NetBeans 集成开发环境中添加外部 SVG 编辑器。选择菜单中的“工具”→“选项”。在打开的“选项”窗口中选择“杂项”节点，然后选择“SVG”标签，单击“浏览”按钮在打开的“选择”对话框中选择 Inkscape 的可执行文件的路径，如图 20-81 所示。

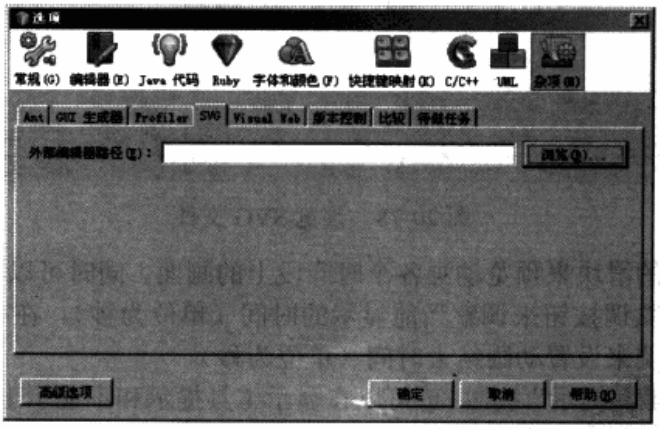


图 20-81 添加外部 SVG 编辑器

在项目窗口右键单击要编辑的 SVG 文件，选择弹出菜单中的“编辑”命令，则会在配置的外部 SVG 编辑器打开选择的 SVG 文件。

20.14.3 导出 SVG 图像和动画

在 NetBeans 集成开发环境中可以将 SVG 图像或动画转换为 PNG 图像。要导出 SVG 图像，在“项目”窗口中右键单击图像节点，然后选择“导出为图像”命令，打开“导出为图像”对话框，如图 20-82 所示。

Download at http://www.pin5i.com/

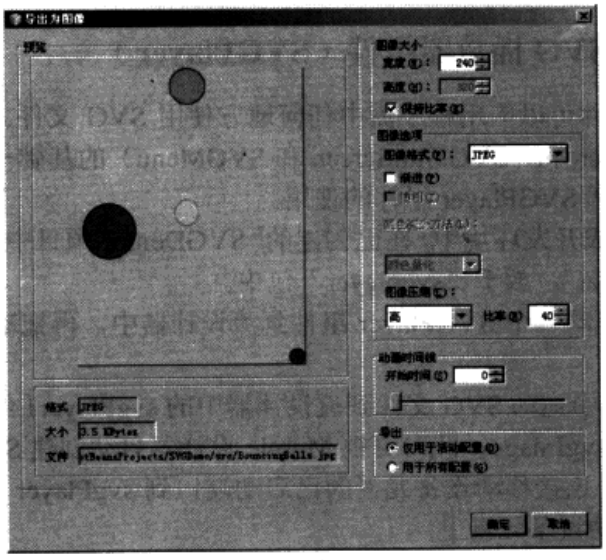


图 20-82 “导出为图像”对话框

可以在该对话框中设置图像大小，并且可以选择将图像导出到项目的活动配置或所有配置。如果选择将图像导出到所有配置，则集成开发环境会自动根据每个配置的屏幕大小创建光栅图像。例如，如果有三种配置，并且每种配置的屏幕大小各不相同，则导出功能会自动创建三个 PNG 图像，它们的大小将与三种屏幕大小对应。

导出 SVG 动画的方法和导出 SVG 图像类似。在“项目”窗口中右键单击图像节点，然后选择弹出菜单中的“将动画导出为图像”命令，打开“正在导出动画”对话框，如图 20-83 所示。

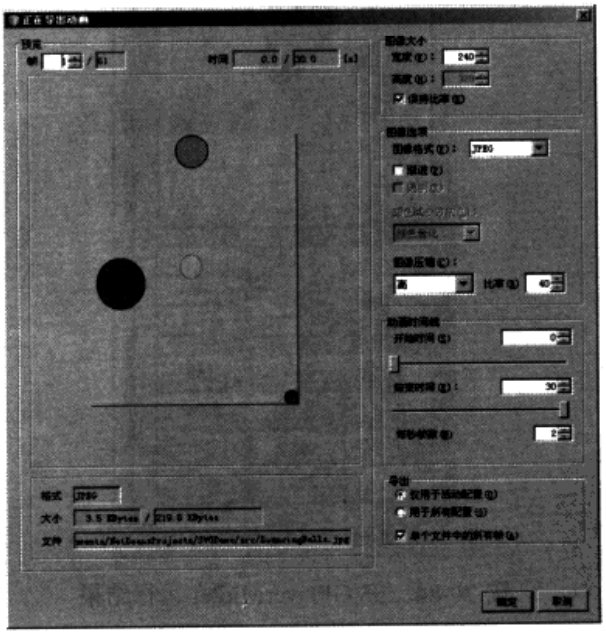


图 20-83 “正在导出动画”对话框

该对话框的设置基本上和导出 SVG 图像一样，这里要说明的是关于“动画时间线”的设置。在“开始时间”微调框中输入或调节出要导出动画的时间段的起始时间（单位为秒），在“结束时间”微调框中输入或调节出要导出动画的时间段的起终止时间（单位为秒）。“每秒帧数”指的是每秒钟内截取的动画帧数。

Download at http://www.pin5i.com/

20.14.4 使用 SVG 播放器组件（SVGPlayer）

使用 SVGPlayer 组件可以应用程序流中任何地方使用 SVG 文件。SVGPlayer 是所有其他 SVG 组件（SVGSplashScreen、SVGWaitScreen 和 SVGMenu）的基础——都是 SVGPlayer 的间接子类。下面以实例讲解 SVGPlayer 组件的使用。

❶ 在 NetBeans 集成开发环境中，在已创建的“SVGDemo”项目中新建一个 Visual MIDlet，名称为“SVGPlayerMidlet”，放于“svgplayer”包中。

❷ 从组件面板中拖动一个 SVGPlayer 组件到流设计器中，再拖动一个退出命令按钮组件到 svgPlayer 组件上。

❸ 从“项目”窗口中拖动 SVG 文件到流设计器中的 svgPlayer1 组件上。此时会导航窗口中的资源节点下会出现 svgImage，此时可视 Mobile 设计器自动创建 SVGImage 组件。

❹ 在流设计器中，连接移动设备元素的已启动端口到 svgPlayer 上，连接 svgPlayer 上的 exitCommand 端口到移动设备元素上。

❺ 按 F6 键运行主项目，在打开的模拟器窗口中启动 SVGPlayerMidlet。可以在模拟器的屏幕上看到动画，如图 20-84 所示。

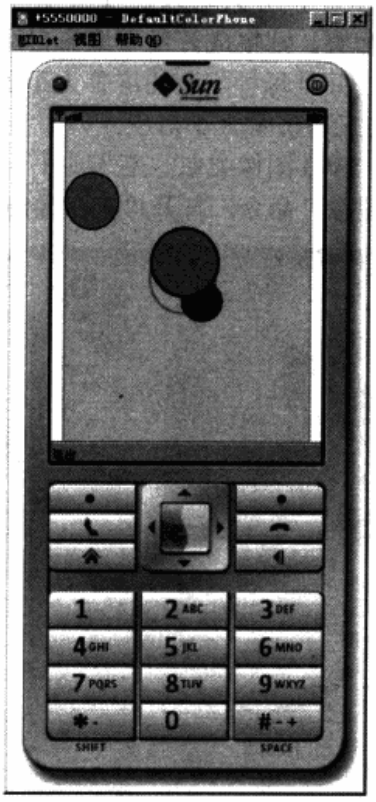


图 20-84 SVGPlayerMidlet 运行结果

SVGPlayer 组件的属性如表 20-26 所示。

表 20-26 SVGPlayer 的属性说明

属 性	说 明
SVG 图像	SVGPlayer 组件中使用的 SVG 图像资源
立即启动动画	当 SVGPlayer 组件显示在屏幕上时是否立即启动动画
动画时间增量	两帧之间的最小时间增加量，单位为秒。其值应该为非负数

Download at http://www.pin5i.com/

续表

属性	说明
停止时重置动画	如果为 true，当 SVGPlayer 组件从屏幕中消失时自动重置动画到开始处；如果为 false，动画不会被重置并且当下次组件被显示时动画自动继续从停止处播放
标题	SVGPlayer 组件窗口的标题
全屏	SVGPlayer 组件是否是全屏模式显示
标题滚动消息栏	与 SVGPlayer 组件关联的滚动消息栏

SVGImage 组件的属性如表 20-27 所示。

表 20-27　SVGImage 的属性说明

属性	说明
外部资源处理程序	获取外部资源的 Java 代码
图像资源路径	SVG 图像文件的路径

20.14.5　使用 SVG 闪屏（SVGSplashScreen 组件）和 SVG 等待屏幕（SVGWaitScreen 组件）

SVGSplashScreen 组件和 SVGWaitScreen 组件的使用方法与 SplashScreen 组件和 WaitScreen 组件的使用方法类似。不同的地方就是 SplashScreen 组件与 WaitScreen 组件使用的是 Image 图像资源，而 SVGSplashScreen 组件与 SVGWaitScreen 组件使用的是 SVGImage 图像资源。

SVGSplashScreen 组件的属性如表 20-28 所示。

表 20-28　SVGSplashScreen 的属性说明

属性	说明
超时	SVG 闪屏显示的时间，单位为毫秒
允许超时中断	是否允许超时中断。如果选中，则用户可以在 SVG 闪屏超时中断，执行下一步操作，否则用户只能等待超时完毕
关闭操作	当超时时间到或用户中断时所执行的操作
SVG 图像	SVG 闪屏窗口中使用的 SVG 图像或 SVG 动画
立即启动动画	当显示 SVG 闪屏窗口时 SVG 动画立即开始播放
动画时间增量	两帧之间的最小时间增加量，单位为秒。其值应该为非负数
停止时重置动画	如果为 true，当 SVGSplashScreen 组件从屏幕中消失时自动重置动画到开始处；如果为 false，动画不会被重置并且当下次组件被显示时动画自动继续从停止处播放
全屏	是否全屏显示
标题	SVG 闪屏窗口的标题，如果全屏属性设置为 true 将不会显示标题

SVGWaitScreen 组件的属性如表 20-29 所示。

表 20-29　SVGWaitScreen 的属性说明

属性	说明
任务	SVG 等待屏幕所执行的任务
成功操作	任务成功时的操作
失败操作	任务失败时的操作
SVG 图像	SVG 等待屏幕中使用的 SVG 图像或 SVG 动画

Download at http://www.pin5i.com/

续表

属 性	说 明
立即启动动画	当显示 SVG 等待屏幕时 SVG 动画立即开始播放
动画时间增量	两帧之间的最小时间增加量，单位为秒。其值应该为非负数
停止时重置动画	如果为 true，当 SVGWaitScreen 组件从屏幕中消失时自动重置动画到开始处；如果为 false，动画不会被重置并且当下次组件被显示时动画自动继续从停止处播放
全屏	是否全屏显示
标题	SVG 等待屏幕的标题，如果全屏属性设置为 true 将不会显示标题

20.14.6 使用 SVG 菜单组件（SVGMenu）

在使用 SVG 图形作为背景和单个菜单项时，并不限制使用图形和动画菜单。在可视 Mobile 设计器中，可以将已创建的菜单添加到应用程序流中，或者先添加菜单背景，然后再将单独的菜单项添加到菜单中。

可以按照下面的两种方式在菜单中创建菜单项：

如果拖放的图像所包含的元素（表示菜单项）ID 是以“menuItem_”字符串开头的，则集成开发环境会自动识别这些元素，并将其作为菜单项添加到组件中（按字母顺序排序）。

如果 SVG 图像不包含 ID 是以“menuItem_”字符串开头的元素，则可以添加表示单个菜单项的元素的名称，只需将组件面板中的 SVGMenu 元素组件拖放至 SVGMenu 组件上即可。这样，就会将 SVGMenu 元素组件添加到 SVGMenu 中，并且在流设计器中可以看到名为“menuItem_”的新可用端口。之后，可以重命名该端口，以便使其与 SVG 文件中的元素 ID 相对应。

下面讲解一个综合实例，其中用到了所有 SVG 组件。

❶ 将预先准备好的 svg 图像文件复制到项目的 src 文件夹中。

❷ 在 SVGDemo 项目中新建一个 Visual MIDlet，名称为“SVGMenuMidlet”，放于“menu”包中。

❸ 从组件面板中拖动一个 SVGSplashScreen 到流设计器中。从项目窗口中拖动 splashscreen.svg 到 svgSplashScreen 上。

❹ 拖动一个 SVGMenu 到流设计器中，将 menu.svg 拖动到 svgMenu 上。这时会看到 svgMenu 组件上出现四个菜单元素的端口。

❺ 分别拖动一个 SVGWaitScreen 和一个 SVGPlayer 到流设计器中。将 waitscreen.svg 拖动到 svgWaitScreen 上，将 halloween.svg 拖动到 SVGPlayer 上。

❻ 选中在导航窗口中资源类别下的 task 节点，在属性窗口中设置其“可执行的代码”属性。转到源代码视图修改 execute 方法的代码如下所示：

```
public SimpleCancellableTask getTask() {
    if (task == null) {
        // write pre-init user code here
        task = new SimpleCancellableTask();
        task.setExecutable(new org.netbeans.microedition.util.Executable()
            {
            public void execute() throws Exception {
                // write task-execution user code here
                Thread.sleep(3000);
            }
        });
        // write post-init user code here
    }
    return task;
}
```

Download at http://www.pin5i.com/

❼ 拖动两个 SVGPlayer 到流设计器中，将 logoAnim3.svg 拖动到 svgPlayer1 上，将 about.svg 拖动到 svgPlayer2 上。

❽ 分别拖动三个 Back Command 命令按钮到三个 SVGPlayer 组件上。

❾ 用鼠标连接移动设备元素的已启动端口到 svgSplashScreen 上。连接 svgSplashScreen 的关闭端口到 svgMenu 上。

❿ 连接 svgMenu 的 menuItem_0_download 端口到 svgWaitScreen 上，连接 svgWaitScreen 的成功端口到 svgPlayer 上，连接 svgPlayer 的后退端口到 svgMenu 上。

⓫ 连接 svgMenu 的 menuItem_1_animate 端口到 svgPlayer1 上，连接 SVGPlayer1 的后退端口到 svgMenu 上。

⓬ 连接 svgMenu 的 menuItem_2_about 端口到 SVGPlayer2 上，连接 SVGPlayer2 的后退端口到 svgMenu 上。

⓭ 连接 svgMenu 的 menuItem_3_exit 端口到移动设备元素上。流设计器中组件连接的最终结果如图 20-85 所示。

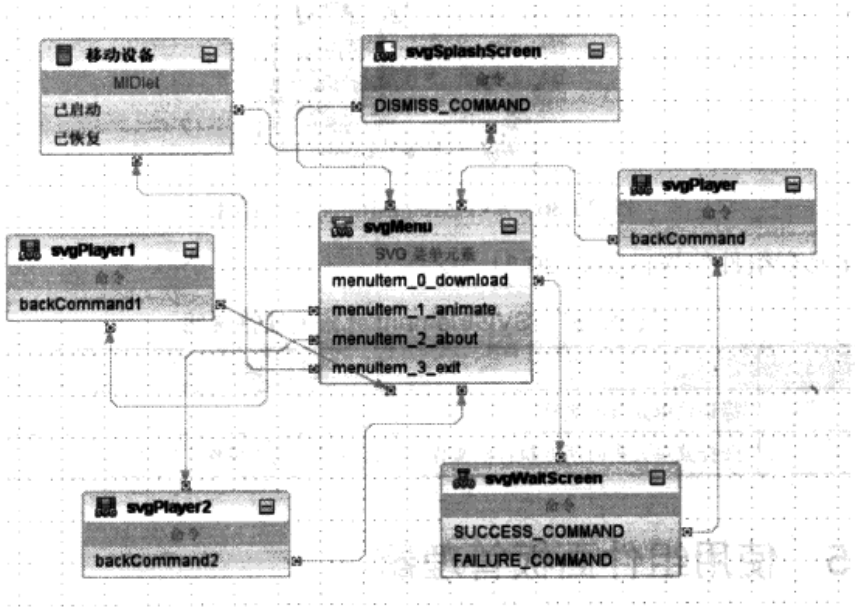


图 20-85 SVGMenuMidlet 最终流设计结果

⓮ 按 F6 键运行主项目，在打开的模拟器窗口中启动 SVGMenuMidlet。SVGMenu 菜单的效果如图 20-86 所示，按上下左右方向键可以在各个菜单项之间切换。

SVGMenu 组件的属性如表 20-30 所示。

表 20-30 SVGMenu 的属性说明

属 性	说 明
SVG 图像	SVG 菜单中使用的 SVG 图像或 SVG 动画
立即启动动画	当显示 SVG 菜单时 SVG 动画立即开始播放
动画时间增量	两帧之间的最小时间增加量，单位为秒。其值应该为非负数
停止时重置动画	如果为 true，当 SVGMenu 组件从屏幕中消失时自动重置动画到开始处；如果为 false，动画不会被重置并且当下次组件被显示时动画自动继续从停止处播放
全屏	是否全屏显示
标题	SVG 菜单的标题，如果全屏属性设置为 true 将不会显示标题

Download at http://www.pin5i.com/



图 20-86 SVGMenuMidlet 运行结果

SVGMenu 元素组件的属性如表 20-31 所示。

表 20-31 SVGElement 元素的属性说明

属 性	说 明
字符串	标识菜单元素的名称
操作	选择菜单元素时所要执行的操作，同命令按钮的“操作”属性

20.15 使用组件面板管理器

可以使用组件面板管理器在可视 Mobile 设计器的组件面板中添加定制组件。在组件面板中添加了定制组件后，可以在设计应用程序时将这些组件拖放到流设计器或屏幕设计器中。

要打开组件面板管理器，请在组件面板中单击鼠标右键，然后选择“组件面板管理器”，打开“组件面板管理器”对话框，如图 20-87 所示。

“组件面板内容”列表中显示了可用于项目的所有组件，处于选中状态的组件将会显示在组件面板中。可以单击“上移”和“下移”按钮来调整选定组件的顺序。可以单击“删除”按钮删除选定的“组件”。单击“新建类别”按钮在打开的“新建组件面板类别”对话框中输入类别的名称创建新的类别。

要将定制组件添加到组件面板中，首选确保要使用的类位于项目的类路径中，然后单击“从项目添加”按钮，打开“选择项目”对话框，如图 20-88 所示。

“选择项目”下拉列表框中列出所有已打开的项目，从中选择包含要添加到组件面板中的对象的项目。单击“下一步”按钮，打开“选择类”对话框，如图 20-89 所示。

从“找到的 MIDP 类”列表中选择要添加的任何对象。该列表显示了选定的项目类路径中所有继承了显示功能类的类或元素。如果选中“显示库中的组件”复选框，可以添加项目的库和资源中的类组件。

Download at http://www.pin5i.com/

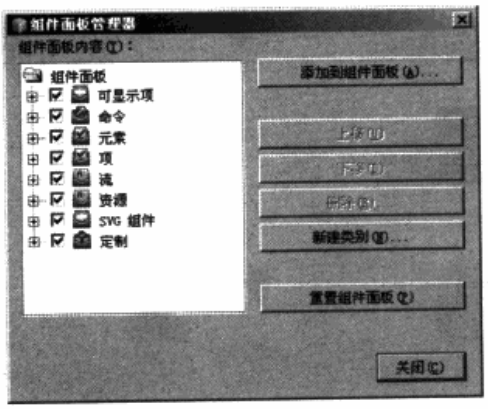


图 20-87 “组件面板管理器”对话框

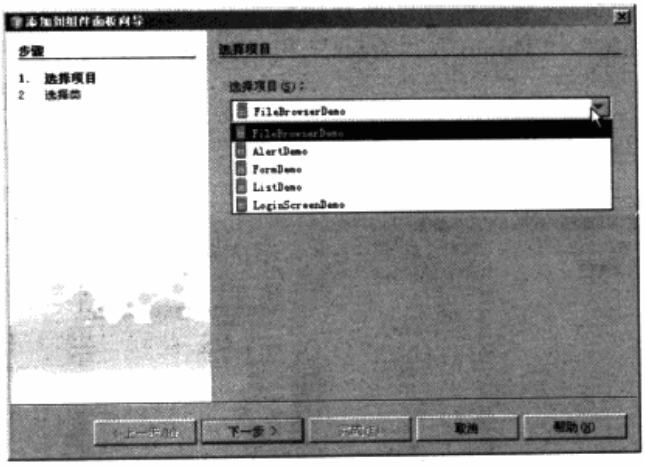


图 20-88 “选择项目”对话框

单击“完成”按钮。在“组件面板管理器”对话框中单击“关闭”按钮，在组件面板中即可看到刚刚添加进来的组件。

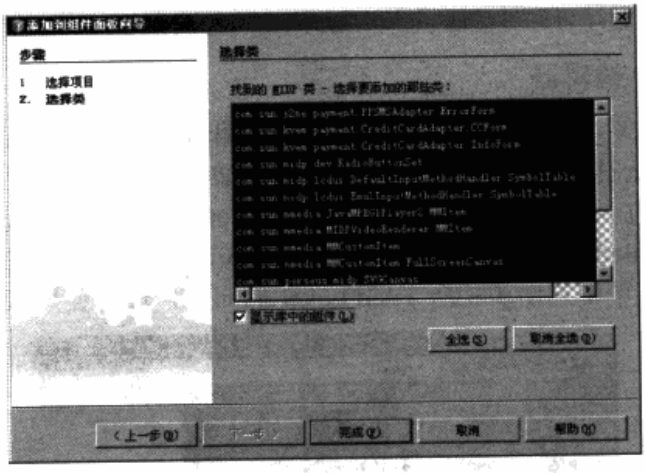


图 20-89 “选择类”对话框

20.16 使用 Mobile Game Builder

游戏生成器（Game Builder）是 NetBeans 6.0 中新增加的特性，其目的在于提供游戏开发的可视化环境，简化开发步骤，从而提高游戏开发的效率。下面首先以 NetBeans IDE 中自带的实例来介绍 Game Builder 的各个组成部分。

20.16.1 创建 Game Builder 样例项目

❶ 在 NetBeans 集成开发环境中新建一个项目（快捷键为 Ctrl + Shift + N），在“新建项目”对话框中，在类别列表中依次选择“样例”→“Mobility”→“MIDP”，在“项目”列表中选择“通过游戏生成器创建的简单游戏”，单击“下一步”按钮，如图 20-90 所示。

❷ 在“名称和位置”对话框中输入项目名称“GameBuilderDemo”并选择项目保存的位置，选中“设置为主项目”复选框，单击“确定”按钮。

❸ 打开 GameDesign.java 文件，则 Game Builder 中打开该游戏设计，如图 20-91 所示。从图中可以看出，一个游戏设计包含三部分内容：场景（Scene）、平铺层（Tiled Layer）和动画层

Download at http://www.pin5i.com/

（Sprite）。对于创建的游戏设计来说包含一个名为“Forest”的场景，四个平铺层，名称分别为“Base”、“Things”、“Trees”和“Water”，两个名称分别为“Karel”和“Thomas”的动画层。

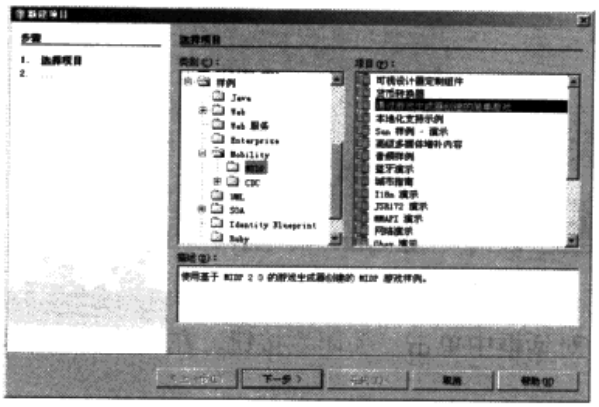


图 20-90 “新建项目”对话框

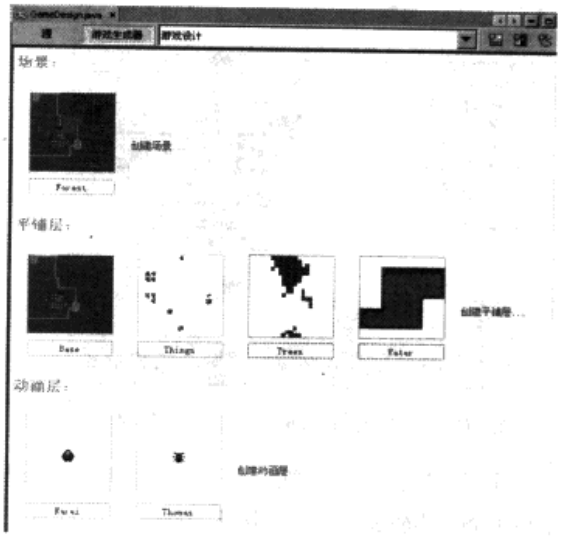


图 20-91 Game Builder 的游戏设计

1. Game Builder 编辑器

用鼠标单击某个元素，可以将其在 Game Builder 编辑器中打开，此处单击 Scene 类别下的“Forest”。Game Builder 中会显示“Forest”的内容，如图 20-92 所示。一个 Scene 是由一个或多个 Tiled Layer 组成的，当鼠标滑动到某个 Tiled Layer 上时，会显示该 Tiled Layer 的边框。用鼠标单击会选中该 Tiled Layer，用鼠标拖动可以改变该 Tiled Layer 在 Scene 中的位置。单击鼠标右键，在弹出的菜单中列出了相应 Tiled Layer 的命令，如改变层次、编辑、删除等。

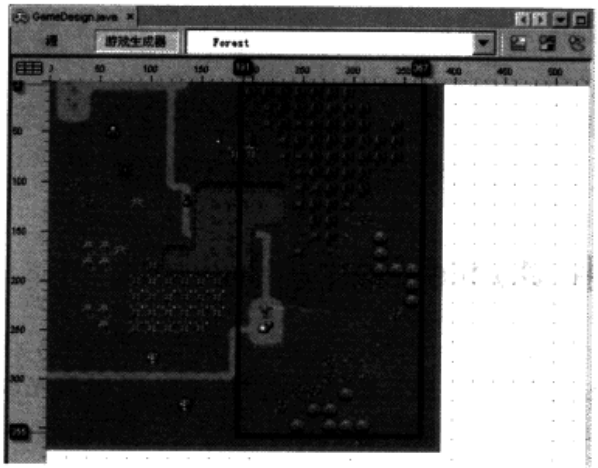


图 20-92 在 Game Builder 中打开“Forest”

在 Game Builder 编辑器面板的顶部是一个工具栏，下面介绍其使用方法。

- “源”按钮：转到源代码视图，在源代码编译器中编辑文件的代码。
- “游戏生成器”按钮：转到图像界面视图，可视化编辑游戏设计。可以 Ctrl + Alt + Page Up 快捷键或 Ctrl + Alt + Page Down 快捷键在源代码视图和游戏生成器视图之间切换。
- 下拉列表框：在列表框中列表出所有已经创建的游戏组件，可以选择相应组件在其之间进行导航。
- “创建新的场景”按钮 ：创建一个新的空场景。

Download at http://www.pin5i.com/

- “创建新的平铺层”按钮：打开“新建平铺层”对话框，创建新的平铺层。
- “创建新的动画层”按钮：打开“创建新的动画层”对话框，创建新的动画层。

2. 导航窗口

在集成开发环境的左侧底部是 Game Builder 的导航窗口。导航窗口的内容会随着 Game Builder 编辑器中打开的内容不同而有所变化。当在编辑器窗口打开一个 Scene 时，其相应地列出所有已经添加到该 Scene 中的 Tiled Layer 以及其属性。当“Forest”在 Game Builder 编辑器打开时，导航窗口显示的内容如图 20-93 所示。

导航窗口中的各部分所表示的内容说明如下。

- Type：图层的类型，用 Tiled Layer 或 Sprite 的图标表示。
- Z：排列顺序，表示哪个图层在顶部哪个图层在底部，0 表示顶部的图层。
- View：图层是否显示，用鼠标单击相应图层 View 下的图标可以在显示与隐藏之间切换。
- Lock：图层是否锁定。如果图层被锁定，则其将不能被移动。
- Name：图层的名称。
- X：图像相对场景的横坐标。
- Y：图像相对场景的纵坐标。

打开 Game Builder 编辑器工具栏上的下拉列表，选择“Water”。因为“Water”是一个 Tiled Layer，导航窗口中显示的内容如图 20-94 所示。窗口中显示 Tiled Layer 的缩略图，并显示其行数和列数等信息。

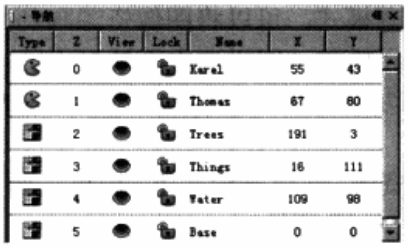


图 20-93 导航窗口

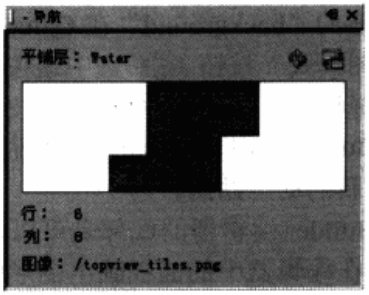


图 20-94 导航窗口

20.16.2 创建场景和平铺层

新建一个 MIDP 应用程序，项目名称为“MyGameBuilder”。在项目新建一个“Visual Game Design”文件，名称为“GameDesign”，放于“demo”包下。将准备好的 PNG 图像文件复制到项目的 src 文件夹中。

1. 创建场景

打开 GameDesign.java 文件，此时 Game Design 中没有任何内容。在 Game Builder 编辑器中单击 Scene 类别下的“创建新的场景”按钮，在打开的“新建场景”对话框中输入场景的名称，如图 20-95 所示。单击“确定”按钮，新建的名为“Desert”的场景就会在 Game Builder 编辑器中打开。

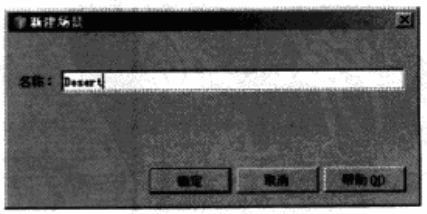


图 20-95 新建场景

2. 创建平铺层

❶ 单击 Game Builder 编辑器工具栏上的“创建新的平铺层”按钮，打开“新建平铺层”对话框。在“平铺层名称”文本框中输入平铺层的名称，此处输入“background”。在“选择图

Download at http://www.pin5i.com/

像”列表框中选择图像文件，调整贴图宽度和贴图高度的滑杆来调整贴图的大小，如图 20-96 所示。

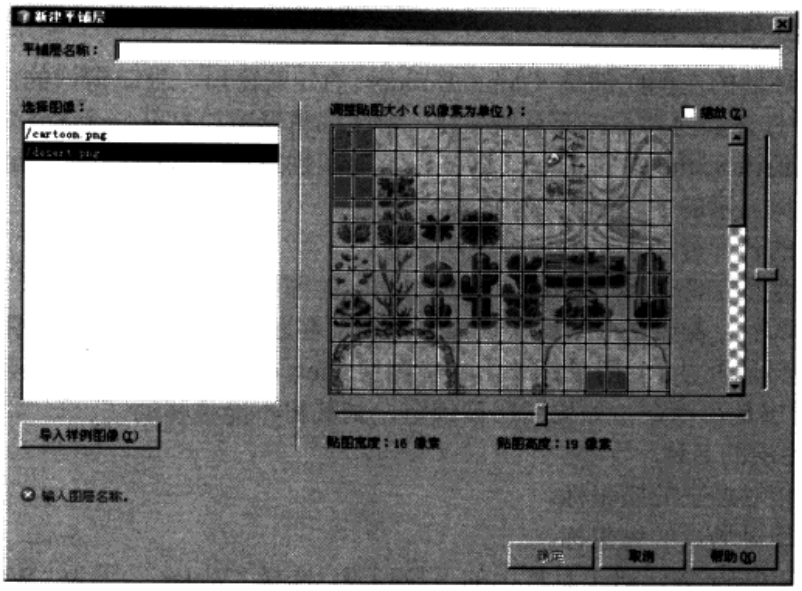


图 20-96 “新建平铺层”对话框

可以选中“缩放”复选框放大图像，调整合适后单击“确定”按钮。

❷ 新建的“background”在 Game Builder 编辑器中打开，其工具栏中有两个按钮。

- “绘图模式”按钮：选中该按钮，进行绘画模式，可以将图像资源画到平铺层中。
- “选择模式”按钮：选中该按钮，进入选择模式，用于选中平铺层中的内容。

Game Builder 编辑器右侧列表出所有已创建的动态贴图，可以单击列表上部的“新建动态贴图”创建新的动态贴图。

Game Builder 编辑器的底部是可以使用的图像资源。在绘画模式下，用单击选中某个贴图，然后用鼠标在编辑器中的面板画出图形即可，如图 20-97 所示。

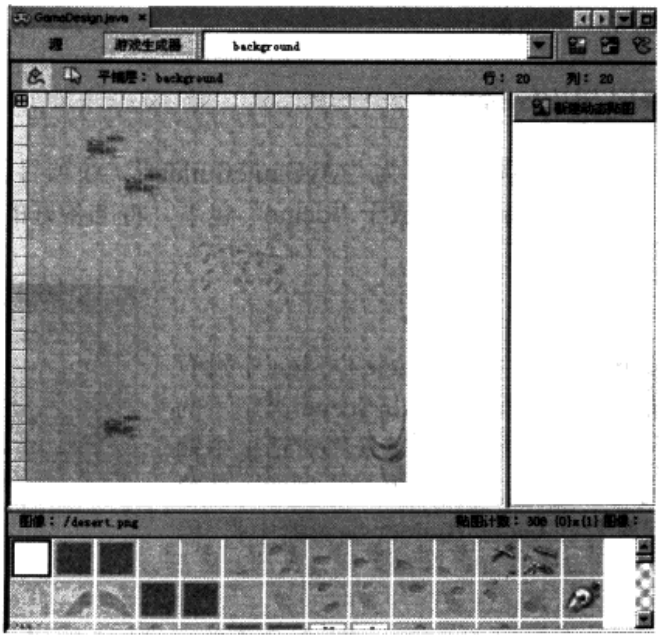


图 20-97 编辑平铺层

Download at http://www.pin5i.com/

当鼠标移动到画布范围之外，Game Builder 会自动添加行或列。可以在某行或列上单击鼠标右键，在弹出的命令菜单中选择添加或删除行和列。

如果画布的大小远远大于所画图形的大小，可以在画布上单击鼠标右键，选择弹出菜单中的“调整到适当大小”命令。

❸ 用同样的方法再创建三个平铺层，其名称分别为“tree”、“flowser”和“trunk”，分别表示树、花和树干，读者可以根据自身的喜好设计。

❹ 完成平铺层的创建之后，就可以将其添加到场景中。在 Game Builder 编辑器工具栏上的下拉列表框中选择“Desert”，则 Desert 场景在 Game Builder 编辑器中打开，此时其还没有任何内容。

❺ 单击鼠标右键，选择弹出菜单中的“添加平铺层”→“background”命令，这样就将 background 添加到 Desert 中了。选中 background，在导航窗口中修改其 X 坐标和 Y 坐标的值都为 0。

❻ 用同样的方法分别将 tree、flowser 和 trunk 添加到 Desert 中，Desert 的最终结果可能如图 20-98 所示。

20.16.3 创建动画层

创建完平铺层之后，下面创建用于表示人物的 Sprite。

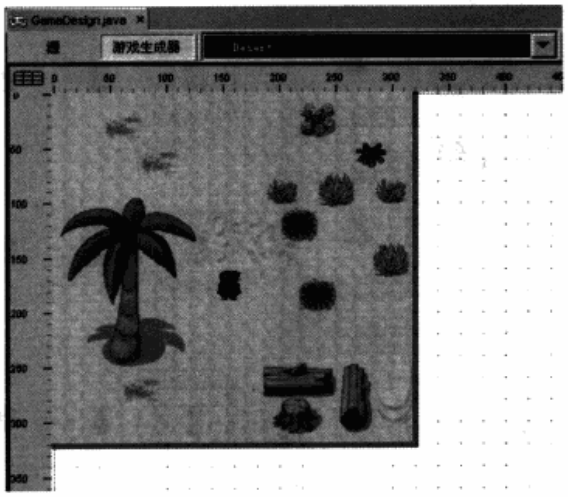


图 20-98 Desert 最终设计结果

❶ 单击 Game Builder 编辑器工具栏上的“创建新的动画图”按钮，打开“创建新的动画层”对话框。在“动画层名称”文本框中输入动画层的名称，此处输入“hero”。在“选择图像”列表框中选择图像文件，调整贴图宽度和贴图高度的滑杆来调整贴图的大小，如图 20-99 所示。

可以选中“缩放”复选框放大图像，调整合适后单击“确定”按钮。

❷ 新建的动画层在 Game Builder 编辑器中打开。默认情况下，Game Builder 会自动创建一个包含 5 帧的名称为“heroseq001”的 Sequence。在导航窗口中修改其名称为“HeroWalkUp”，在 Game Builder 编辑器中用鼠标右键单击第 5 帧，在弹出的菜单中选择“删除序列”命令，删除第 5 帧，使用同样的方法删除第 4 帧。

❸ 下面修改帧的内容，用鼠标拖动索引为 1（第 2 个位置）的图像资源到第 2 帧中，如图 20-100 所示。使用同样的方法修改第 3 帧的内容。

修改完成后，可以单击 按钮从前向后播放，也可以单击 按钮从后向前播放，单击 按钮暂停播放。使用延迟微调框可以调整两帧之间的时间间隔。

Download at http://www.pin5i.com/

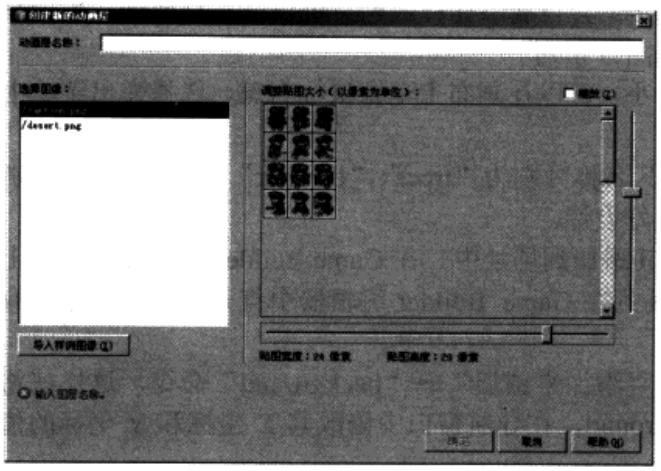


图 20-99 “创建新的动画层”对话框

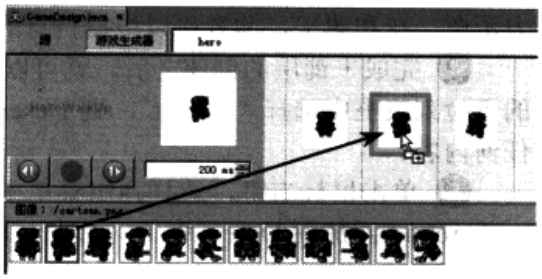


图 20-100 修改帧内容

❹ 在已经创建的序列上单击鼠标右键，在弹出的菜单中选择“创建序列”命令，可以创建新的序列。使用上述的方法再创建三个和 HeroWalkUp 类似的序列，名称分别为“HeroWalkDown”、“HeroWalkLeft”和“HeroWalkRight”，分别表示向下走、向左走和向右走。这四个序列的最终设计结果如图 20-101 所示。

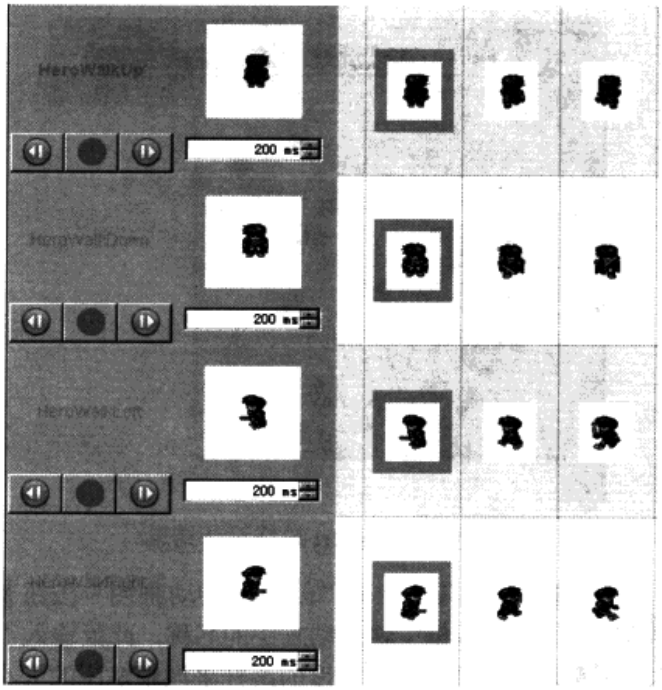


图 20-101 Hero 最终设计结果

❺ Hero 创建完成后，可以将其添加到 Desert 中。转到 Desert，在编辑器中单击鼠标右键，选择弹出菜单中的“添加动画层”→“hero”命令，并 hero 移动到适当的位置。

至此，游戏的设计已经完成，但其还不能运行，还需要创建 GameCanvas 和 MIDlet。

20.16.4 创建 GameCanvas 和 MIDlet

在 demo 包中新建一个 Java 类，名称为“DemoGameCanvas”。参考前面示例文件的代码，修改 DemoGameCanvas.java 文件的代码，由于篇幅的限制，此处不列出代码，请读者参考相关

Download at http://www.pin5i.com/

源代码文件。

在 demo 包中新建一个 MIDlet，名称为“GameMidlet”，修改其文件的代码如下所示：

```
package demo;

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class GameMidlet extends MIDlet {

    private DemoGameCanvas gameCanvas;
    private Thread thread;
    private Display display;

    public void startApp() {
        this.gameCanvas = new DemoGameCanvas();//创建 DemoGameCanvas 实例
        this.thread = new Thread(gameCanvas);
        thread.start();//启动线程
        display = Display.getDisplay(this);
        display.setCurrent(gameCanvas);//显示 gameCanvas
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
        this.gameCanvas.stop();
    }
}
```

至此整个游戏创建完毕，运行 MyGameBuilderDemo。在打开的模拟器中启动 GameMidlet，游戏运行结果如图 20-102 所示。

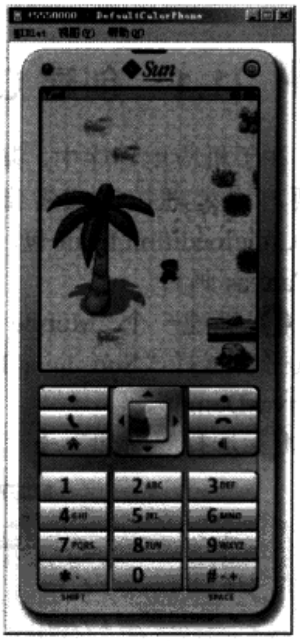


图 20-102 GameMidlet运行结果

20.17 小结

本章主要介绍利用可视 Mobile 设计器以可视化的方式开发 Java ME 应用程序，包括可视 Mobile 设计器的组成及使用，各种组件的用途、使用方法以及属性的设置等内容。通过本章的学习，读者应该掌握以下基本内容：

- 可视 Mobile 设计器的组成及使用方法
- 各种标准 MIDP 组件和 Mobility Pack 提供的各种非标准 MIDP 组件之间的关系
- 各种组件的用途及使用方法
- 如何设置组件属性
- 定制组件面板
- Mobile Game Builder 的使用

Download at http://www.pin5i.com/

第21章

Java ME 开发高级技术

前面两章中介绍了在 NetBeans 集成开发环境中开发 Java ME 程序的基本方法，本章将介绍一些高级技术，包括签名运行、进行混淆、进行单元测试等内容。对于初学者，可以跳过本章继续学习后续章节。

本章主要内容

- 如何利用向导创建画布
- 签名的概念以及具体方法
- 为什么要进行混淆以及进行混淆的方法
- 多配置管理

本章亮点

- 利用 JMUnit 进行单元测试
- 预编译

21.1 创建 Canvas 画布

在手机应用程序中，基于画布的应用程序属于低级用户界面开发，因为在应用程序中用户需要自己处理按键或者指针事件。画布的大多数操作封装在了 Canvas 类中，该类属于 javax.microedition.lcdui 包。当要开发处理低级按键事件或者在屏幕上绘制图形文字时，可以使用 Canvas 画布。

❶ 新建一个“Mobile 应用程序”项目，项目名称为“CanvasDemo”，不创建 Hello MIDlet，仿真平台选择“Sun Java(TM) Wireless Toolkit 2.5.2 for CLDC”。新建一个 MIDlet，名字为“CanvasDemo”。

❷ 选择“新建”→“新建文件”命令，打开“新建文件”对话框，如图 21-1 所示。

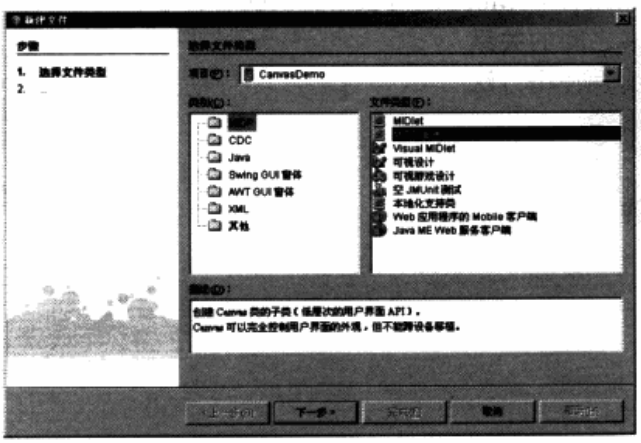


图 21-1 “新建文件”对话框

Download at http://www.pin5i.com/

❸ 在“类别”列表中选择“MIDP”节点，然后在右边的“文件类型”列表中选择“MIDP画布”节点。单击“下一步”按钮，在下面出现的对话框的“MIDP 类名”文本框中输入MIDPCanvas。单击“完成”按钮，创建一个空白的画布，代码如下所示：

```
/*
 * MIDPCanvas.java
 *
 * Created on 2007年11月27日, 上午4:47
 */

import javax.microedition.lcdui.*;

/**
 *
 * @author  Liu Bin
 * @version
 */
public class MIDPCanvas extends Canvas implements CommandListener {
    /**
     * constructor
     */
    public MIDPCanvas() {
        try {
            // Set up this canvas to listen to command events
            setCommandListener(this);
            // Add the Exit command
            addCommand(new Command("Exit", Command.EXIT, 1));
        } catch(Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * paint
     */
    public void paint(Graphics g) {
        g.drawString("Sample Text",0,0,Graphics.TOP|Graphics.LEFT);
    }

    /**
     * Called when a key is pressed.
     */
    protected  void keyPressed(int keyCode) {
    }

    /**
     * Called when a key is released.
     */
    protected  void keyReleased(int keyCode) {
    }

    /**
     * Called when a key is repeated (held down).
     */
    protected  void keyRepeated(int keyCode) {
    }

    /**
     * Called when the pointer is dragged.
     */
    protected  void pointerDragged(int x, int y) {
    }

    /**
     * Called when the pointer is pressed.
     */
```

Download at http://www.pin5i.com/


```
    protected  void pointerPressed(int x, int y) {
    }

    /**
     * Called when the pointer is released.
     */
    protected  void pointerReleased(int x, int y) {
    }

    /**
     * Called when action should be handled
     */
    public void commandAction(Command command, Displayable displayable) {
    }

}
```

❹ 修改 MIDPCanvas 的 paint 方法，使之显示为黑底红字：

```
public void paint(Graphics g) {
    //设置画图颜色为黑色
    g.setColor(0);
    //绘制背景
    g.fillRect(0, 0, getWidth(), getHeight());
    //设置画图颜色为红色
    g.setColor(255, 0, 0);
    //绘制字符串
    g.drawString("Sample Text",0,0,Graphics.TOP|Graphics.LEFT);
}
```

❺ 在 MIDlet CanvasDemo 的 startApp 中添加如下代码，用于显示创建的画布：

```
public void startApp() {
    MIDPCanvas canvas = new MIDPCanvas();
    Display display = Display.getDisplay(this);
    display.setCurrent(canvas);
}
```

❻ 运行项目，显示界面如图 21-2 所示。



图 21-2　画布程序的运行界面

Download at http://www.pin5i.com/

21.2 应用程序的签名运行

在 MIDP 2.0 规范中规定，使用一些和个人信息相关的 API 时（例如，文件系统、电话号码等），必须首先对 MIDlet 进行签名才可以。

21.2.1 签名概述

各个手机制造厂商在实现各自的手机设备时，都有各自不同的实现方法。同样，即使对于同一个品牌的手机，对于不同的移动运营商，也有不同的安全和签名策略。因此，在进行签名前，开发人员需要具体弄清楚签名的策略，以及使用正确的证书。

NetBeans 中提供了为应用程序签名的功能，可以方便地为应用程序进行签名。下面几种情况，需要对 MIDlet 先进行签名然后再运行。

- 使用一些受保护的 API，例如文件系统 API、电话号码本等。
- 不希望出现一些提示信息时。例如联网时，一半会出现进行网络连接的提示对话框，用户确定后才能继续进行网络连接。在 MIDlet 签名后，可以使网络连接的对话框不出现。

在源代码目录中提供了一个名为 ListRoots 的项目，用于显示当前手机文件系统中的根目录。在模拟器上运行该程序的时候，会首先出现如图 21-3 所示的提示对话框，当选择 Yes 命令按钮后，才会出现如图 21-4 所示的文件系统。



图 21-3 访问文件系统的提示对话框

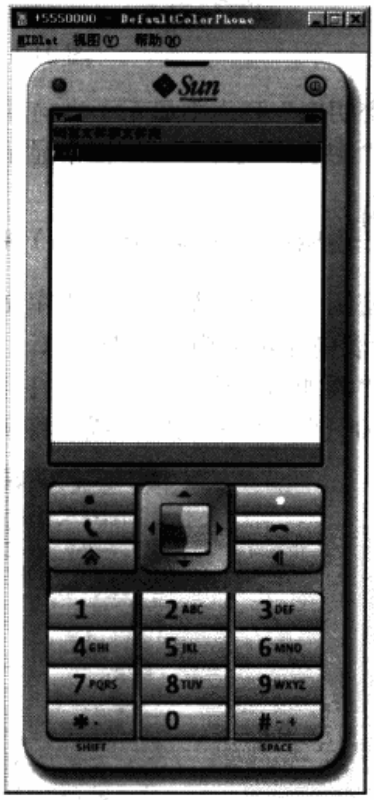


图 21-4 显示文件系统

如果对程序进行签名，使其运行在安全的域上（例如设备制造商域），那么将不会出现如图 21-3 所示的提示对话框。

Download at http://www.pin5i.com/

注意：虽然在模拟器上，不签名的程序也可以访问文件系统，但是在手机上运行时，将会出现安全异常。

21.2.2 NetBeans 的安全管理器

NetBeans 使用安全管理器来管理 MIDP 项目的安全密钥库。在 NetBeans 集成开发环境中选择“工具”→“密钥库”命令，可以打开“密钥库管理器”对话框，如图 21-5 所示。

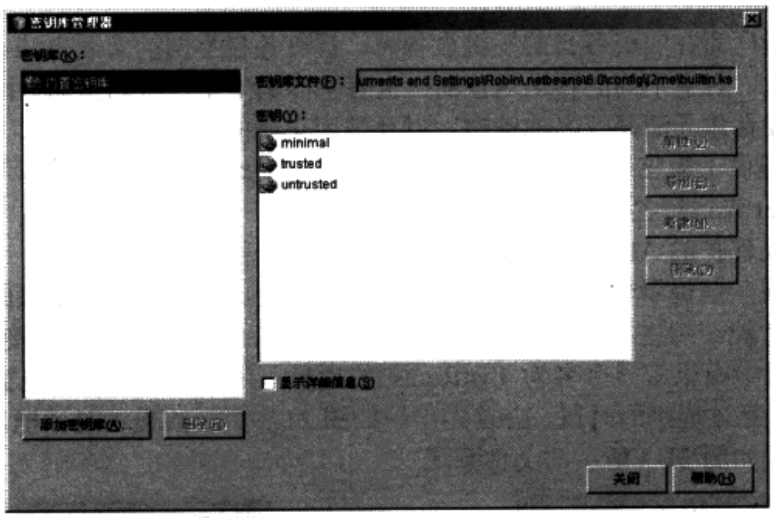


图 21-5 “密钥库管理器”对话框

在“密钥库”列表中显示了当前项目的所有密钥库。从该列表中选择要编辑的密钥库，会在右边的“密钥库文件”文本框中显示密钥文件路径。

说明：密钥库文件（.ks）用于存储签名 MIDlet 套件时使用的密钥对。

在“密钥”列表中显示该密钥库文件中包含的密钥。选中“显示详细信息”复选框，会在“密钥”列表中显示密钥的详细信息。

单击“添加密钥库”按钮，会打开“添加密钥库”对话框，如图 21-6 所示。使用该对话框可以添加一个新的密钥库，或者将现有的密钥库导入到项目中。

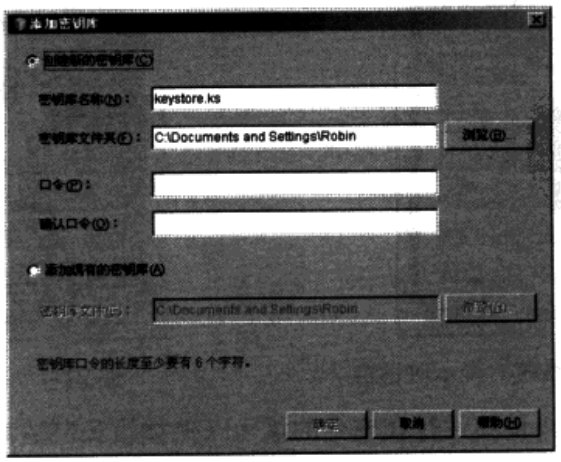


图 21-6 “添加密钥库”对话框

Download at http://www.pin5i.com/

选中“创建新的密钥库”单选按钮，将创建一个新的密钥库文件。在“密钥库名称”文本框中输入密钥库文件的名字；在“密钥库文件夹”文本框中输入密钥库文件的路径。在“口令”文本框中输入用于解锁密钥库文件（以便使用该文件）的口令。口令的长度至少为 6 个字符。在“确认口令”文本框中重新键入口令，确保两次输入的口令一致。

若选中“添加现有的密钥库”单选按钮，可以添加系统中已有的密钥库文件。单击“浏览”按钮以查找该密钥库文件。

21.2.3 MIDP 项目签名

在“项目”窗口中右键单击 MIDP 项目节点，选择弹出菜单中的“属性”命令，打开项目属性对话框。单击“生成”→“签名”节点，在这里可以为 MIDlet 套件设置安全性，如图 21-7 所示。

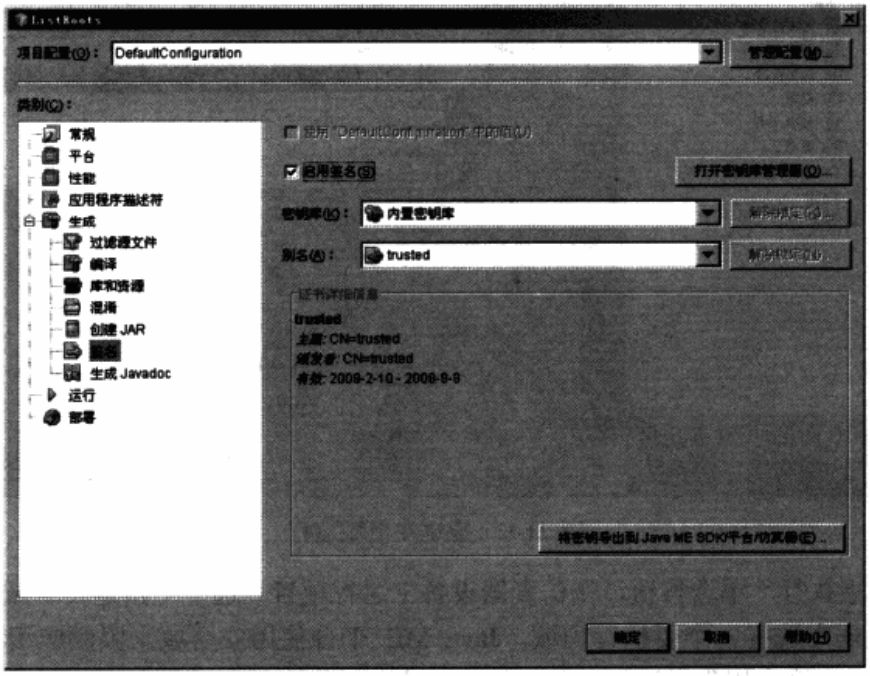


图 21-7 MIDP 项目签名

选中“启用签名”复选框，为 MIDlet 套件启用签名。如果取消选中，则禁用该页面。单击“打开密钥库管理器”按钮，可以打开 NetBeans 的安全管理器。

在“密钥库”下拉列表框中列出了所有可用的密钥库。通过单击“解除锁定”按钮，可以解除锁定要使用的密钥库。

“别名”下拉列表框列出了密钥库中的密钥对指定的名称。通过单击“解除锁定”按钮，可以解除锁定要使用的密钥对。

“证书详细信息”分组框中显示了为密钥对分配的证书的相关信息。单击“将密钥导出到 Java ME SDK/平台/仿真器”按钮，可以将密钥导出到用于测试 MIDlet 的仿真器平台。



说明：如果正在设置项目配置的属性，而此项目配置不是默认配置（DefaultConfiguration），则必须取消选中页面顶部的“使用'DefaultConfiguration'中的值”复选框，才能定制这些选项。

Download at http://www.pin5i.com/

21.2.4 指定安全域运行

在执行 MIDlet 套件的时候，可以选择执行安全域。在“项目”窗口中右键单击 MIDP 项目节点，选择弹出菜单中的“属性”命令，打开项目属性对话框。单击“运行”节点，可以指定运行选项，如图 21-8 所示。

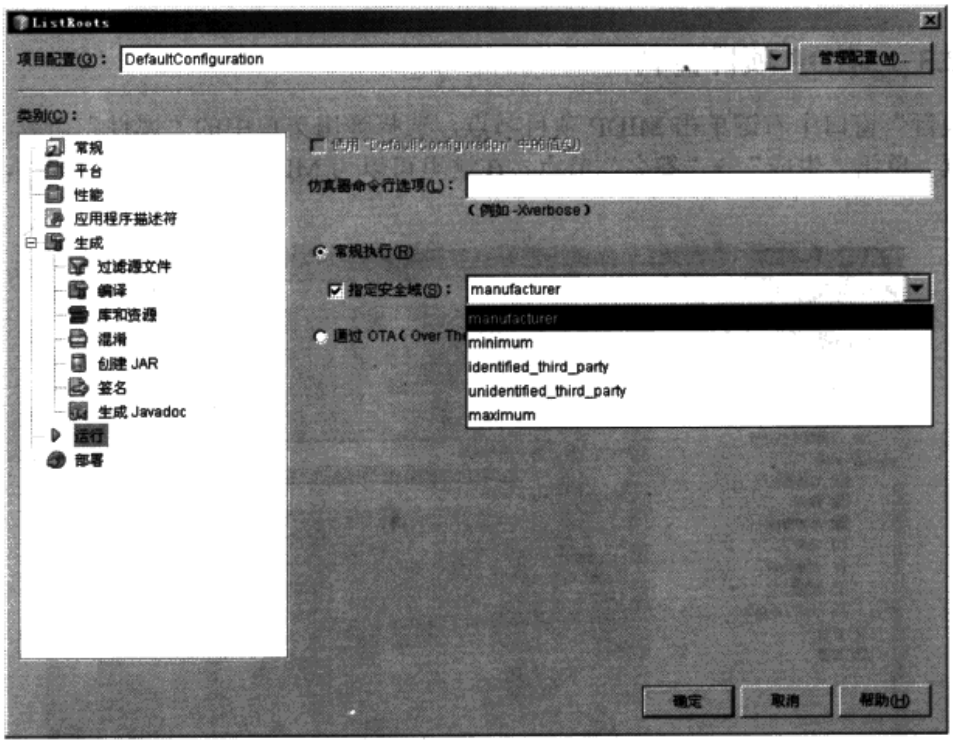


图 21-8 指定安全域运行

选中“常规执行”单选按钮，在仿真器设备上运行项目。选中“指定安全域”复选框，然后在下拉列表框中选择一个要执行的域。Java ME 平台使用安全域来限制应用程序对受保护 API 的访问。在允许应用程序访问受保护的 API 之前，设备使用签名信息查看此应用程序的来源及有效性。

虽然可用的安全域因设备的不同而有所不同，但是 NetBeans 和 Sun Java Wireless Toolkit 均支持以下安全域。

- 可信（Trusted）：具有已使用证书链（可经设备验证）签名且未被篡改的 JAR 文件。
- 不可信（Untrusted）：设备无法信任其 JAR 文件的来源及完整性（例如未签名的 MIDlet 套件）。
- 最低（Minimal）：在其中对受保护的 API 的所有权限（包括访问推功能及网络协议的权限）均被拒绝。
- 最大（Maximal）：相当于“可信”域。

21.3 对应用程序进行混淆处理

Java 语言的一个特性就是跨平台性，因此 Java 语言的编译采用了字节代码格式。也就是说，

Download at http://www.pin5i.com/

源代码中包含的大部分信息仍保留为字节代码格式。这虽然增加了程序的可移植性，但是也带来了安全问题，即用户可以轻松对 Java 应用程序进行反编译或逆向工程获得源代码。

此外，此字节代码格式还会导致文件大小的增加。由于移动设备资源有限，MIDP 应用程序中保存的每个字节都会影响到应用程序的性能。

为了减小 MIDlet 套件的大小，增加反编译的难度，可以对 MIDlet 进行混淆处理。目前有很多混淆工具，例如 Proguard 等。混淆器可以使用各种方法来转换已编译的代码。代码混淆技术包括去除其符号表的对象代码，或更改变量和标识符的名称。

NetBeans 集成开发环境也支持混淆功能。在“项目”窗口中右键单击 MIDP 项目节点，选择弹出菜单中的“属性”命令，打开项目属性对话框。单击“生成”→“混淆”节点，在这里可以对 MIDlet 套件进行混淆设置，如图 21-9 所示。

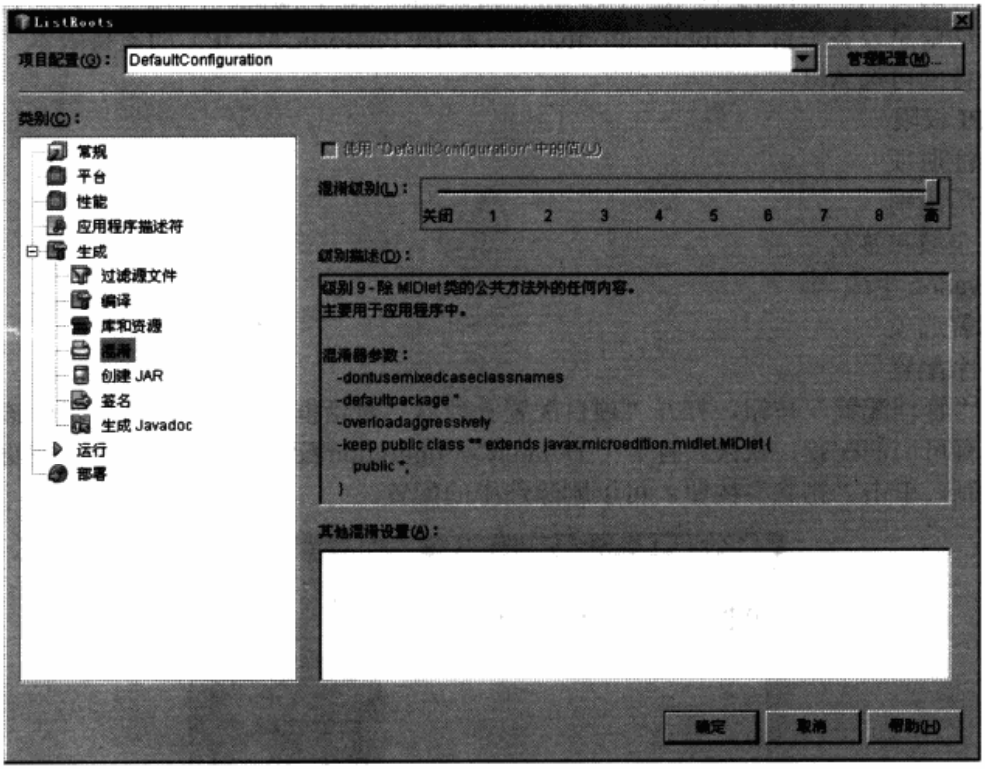


图 21-9 对 MIDlet 套件进行混淆设置

拖动“混淆级别”滑块可以设置混淆的级别，级别越高，混淆的程度也越高。但是，并不是混淆级别越高越好，因为有些手机在实现的时候对混淆器以及混淆的级别都有限制。

可以在“其他混淆设置”文本框中添加其他混淆参数。NetBeans 集成开发环境中包含的默认混淆器为 ProGuard。

21.4 多编译配置以及预编译

众所周知，不同的手机设备可能会有不同的屏幕大小、颜色深度等。这些不同限制了 MIDlet 程序的可移植性，例如虽然一个程序可以在另外一个支持 MIDP 2.0 的手机上运行，但是由于屏幕大小的不同，使得程序界面变得非常难看。为了能够在不同设备上运行，往往需要针对一些手机进行重新设计和编译，生成不同版本的程序。

Download at http://www.pin5i.com/

NetBeans 集成开发环境中提供了一个非常优秀的功能，可以使用多个配置文件。可以使用项目配置为要运行应用程序的每个设备定义执行环境。通过使用项目配置和预处理，可以编写单个应用程序，并为多个设备创建、定制以及部署单独地分发 JAR 文件。



21.4.1 配置管理

在“项目”窗口中右键单击 MIDP 项目节点，选择弹出菜单中的“属性”命令，打开项目属性对话框。在该对话框中可以看到一个“项目配置”下拉列表框，在这里可以选择要使用的配置文件。在项目配置中定义的属性包括：

- 仿真器平台
- 特定的仿真器设备
- 联网的受限设备配置（Connected Limited Device Configuration, CLDC）的支持版本
- 移动信息设备平台（Mobile Information Device Platform, MIDP）的支持版本
- 支持的可选 API
- API 权限
- 推注册项
- 混淆级别
- 安全域和签名
- Javadoc 生成
- 部署选项
- 多个配置

单击“管理配置”按钮，打开“项目配置管理器”对话框，如图 21-10 所示。该对话框中列出了所有可用的配置，默认只有一个 DefaultConfiguration 配置。单击“复制”按钮可以对配置进行复制。单击“删除”按钮，可以删除选中的配置。

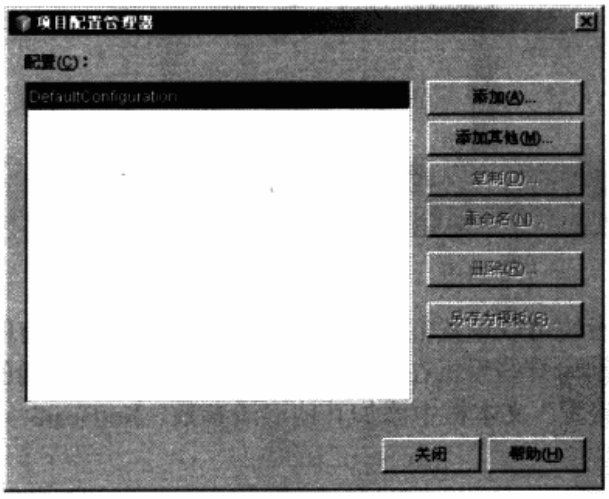


图 21-10 “项目配置管理器”对话框

单击“添加”按钮，会打开如图 21-11 所示的“添加项目配置”对话框，添加一个新的项目配置。在“使用配置模板”下拉列表框中可以选择一些配置模板，然后再进行修改。在“新建配置名称”文本框中可以输入新配置的名字。单击“确定”按钮，会在“项目配置管理器”对话框中添加一个配置。

Download at http://www.pin5i.com/

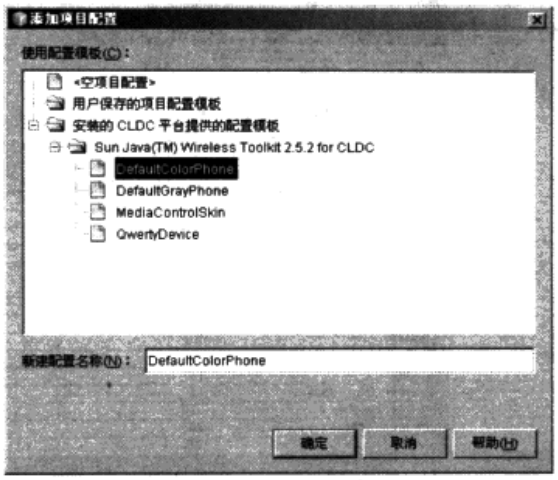


图 21-11 “添加项目配置”对话框

21.4.2 使用更多配置

创建一个名为 MultiConfig 的 MIDP 项目，选择创建 Hello MIDlet。在创建 MIDP 项目时，会出现如图 21-12 所示的对话框。如果要使用多个配置，可以在项目配置模板中通过选中复选框进行选择。

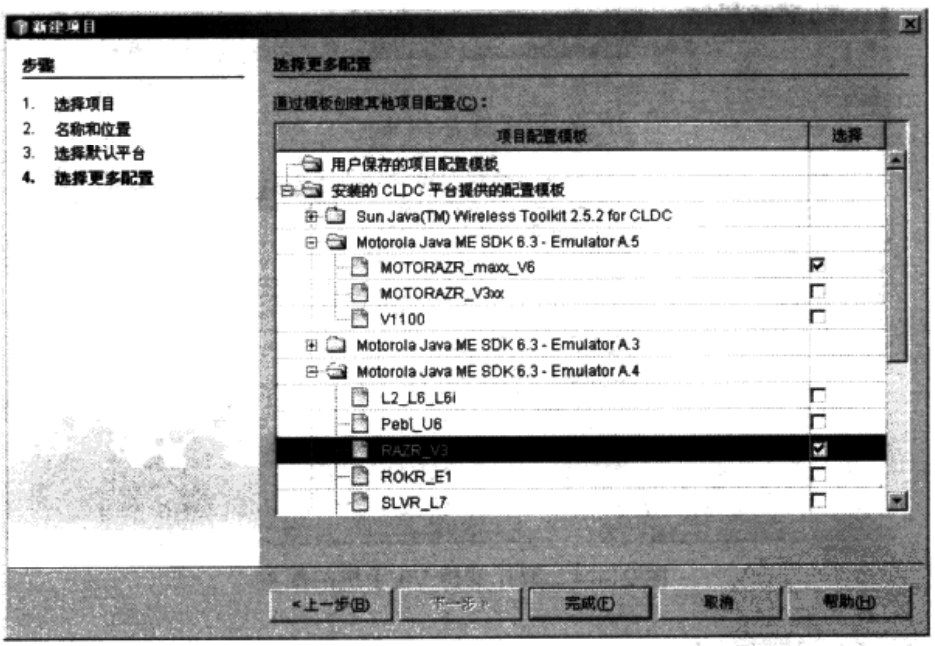


图 21-12 选择更多配置

在“项目”窗口中右键单击 MIDP 项目节点，选择弹出菜单中的“属性”命令，打开项目属性对话框。单击“性能”节点，在“项目配置”下拉列表框选择摩托罗拉 V6 手机的配置，如图 21-13 所示。同时也可以看到如下的参数：

```
ScreenHeight  320
ScreenWidth   240
```

Download at http://www.pin5i.com/

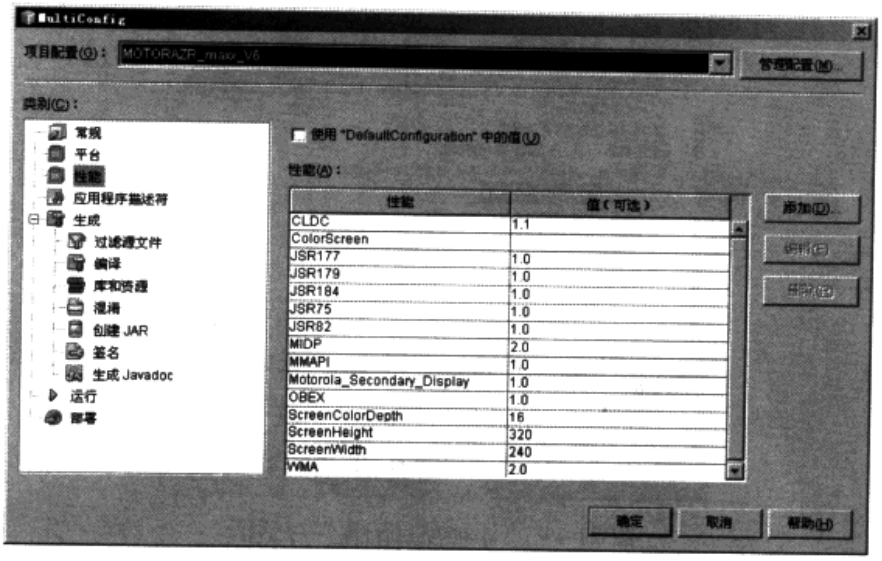


图 21-13 摩托罗拉 V6 手机的配置

在“项目配置”下拉列表框中选择摩托罗拉 V3 手机的配置，如图 21-14 所示。同时也可以看到如下的参数：

```
ScreenHeight  220
ScreenWidth   176
```

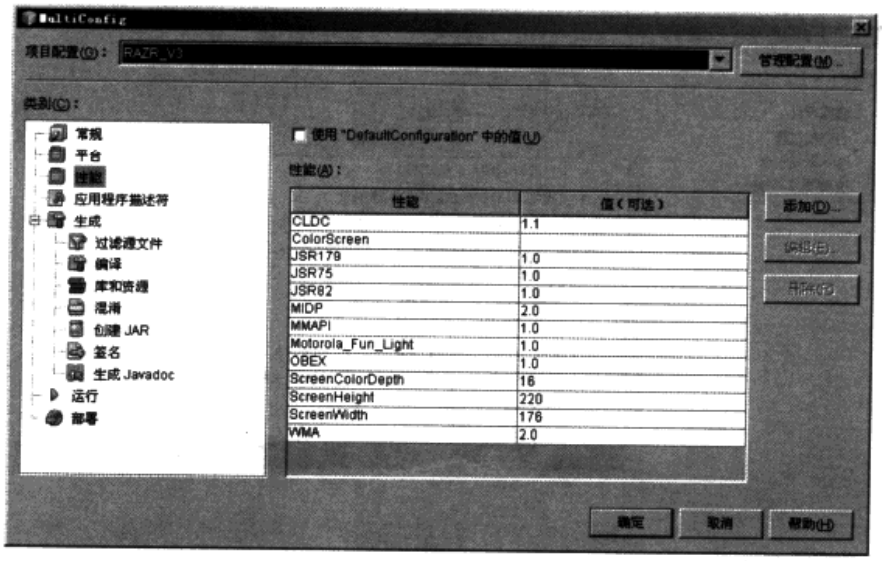


图 21-14 摩托罗拉 V3 手机的配置



21.4.3 使用预编译

大多数的预处理是由预处理程序指令控制的。这些指令是在源代码中指定的，只需创建以//#字符序列开头的注释行，然后直接在字符序列后面跟随指令即可。例如，//#ifdef。

类似于 XML 语言，预处理程序指令的格式必须正确无误。即它们必须包含开始指令和结束指令。例如，当块以//#if 指令开头时，它必须以//#endif 指令结束。可以对块进行嵌套，这意味着 if/elif/else/endif 块内可以包含任意数量的其他 if/elif/else/endif 块。

下面假定程序中对屏幕的大小敏感，需要对不同的屏幕进行不同的处理。在源代码编辑器中打开 MIDlet，单击 startApp 方法中的空白处，选择弹出菜单中的“预处理程序块”→“创建

Download at http://www.pin5i.com/

If/Else 程序块”命令，创建程序块如图 21-15 所示。



图 21-15 创建 If/Else 程序块

在编译器解析代码之前，预处理操作将修改源文件中的代码。预处理程序根据作为代码块插入代码中的预处理程序指令来修改该代码。在源代码编辑器中，这些代码块是以可视方式标记的，为特定的项目配置或性能生成 JAR 时可以包括（或排除）这些代码块。下面为 V6 和 V3 配置，添加如下的预处理代码：

```
    public void startApp() {
        initialize();
//#if MOTORAZR_maxx_V6
        helloForm.append("\nMOTORAZR_maxx_V6");
//#else
        helloForm.append("\nRAZR_V3");
//#endif
    }
```

当选择使用 V3 的配置时，以上代码自动变为：

```
//#if MOTORAZR_maxx_V6
//#        helloForm.append("\nMOTORAZR_maxx_V6");
//#else
        helloForm.append("\nRAZR_V3");
//#endif
```

当选择使用 V6 的配置时，以上代码自动变为：

```
//#if MOTORAZR_maxx_V6
        helloForm.append("\nMOTORAZR_maxx_V6");
//#else
//#        helloForm.append("\nRAZR_V3");
//#endif
```

生成程序后，在 Windows 中打开项目目录下的 build 子目录，可以看到为两个手机自动生成了两个分发，如图 21-16 所示。

Download at http://www.pin5i.com/

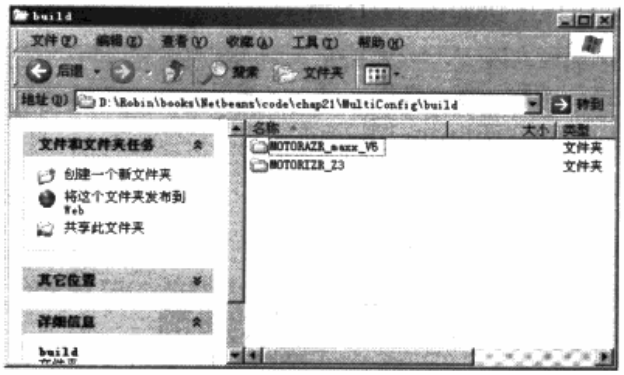


图 21-16 创建的不同分发

当运行程序的时候，在不同的设备上运行，会出现不同的运行结果。图 21-17 是在摩托罗拉 V3 仿真器上的运行界面；图 21-18 是在摩托罗拉 V6 仿真器上的运行界面。



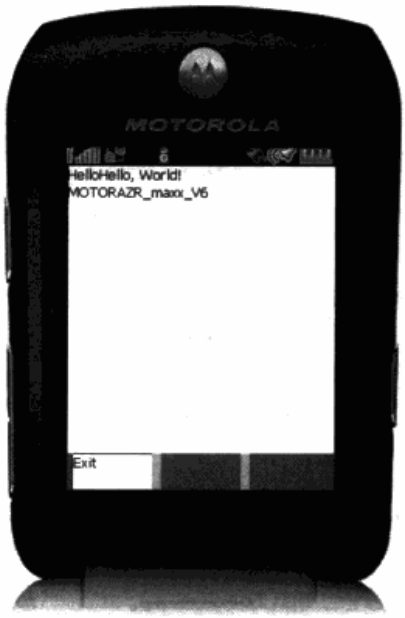


图 21-17 在摩托罗拉 V3 仿真器上的运行界面　　图 21-18 在摩托罗拉 V6 仿真器上的运行界面

21.5 小结

本章主要介绍 Java ME 应用程序中一些高级技术，包括签名运行、多配置和预编译等。其中，多配置是 NetBeans 的一个非常显著的特点。使用该特性，开发人员可以同时对不同的手机设备进行开发。多配置往往都要和预编译一起工作。通过本章的学习，读者应该掌握以下基本内容：

- 如何创建画布
- 签名的基本概念
- 如何对程序进行混淆
- 如何为程序选择多个配置

Download at http://www.pin5i.com/

PART 4

第四篇　NetBeans 组件开发

第 22 章　开发 NetBeans 插件模块

Download at http://www.pin5i.com/

第22章

开发NetBeans插件模块

NetBeans是Sun公司领导的一个成功的开发源码项目。前面几章主要介绍了NetBeans集成开发环境的各种使用和开发方法。本章将介绍NetBeans项目中另外一个产品：NetBeans平台。开发人员可以在NetBeans平台的基础上进行自己的应用程序开发，从而加快程序的开发过程。

本章首先介绍NetBeans集成开发环境和NetBeans平台的区别，然后着重介绍如何开发NetBeans集成开发环境扩展模块，以及开发基于NetBeans平台的富客户端应用程序。本章旨在为NetBeans平台开发的读者指引一条入门的道路。对于不打算进行NetBeans平台开发的读者，只需要阅读本书的第一节，弄清楚NetBeans集成开发环境和NetBeans平台的概念即可。

本章主要内容

- 什么是NetBeans平台及其特点
- 如何创建NetBeans插件模块
- 如何创建操作
- 如何创建模块套件项目

本章亮点

- 如何为操作添加处理代码
- 如何管理NetBeans插件模块项目



22.1 NetBeans IDE和NetBeans平台

目前NetBeans项目中包括两个产品：NetBeans IDE（NetBeans集成开发环境）和NetBeans Platform（NetBeans平台）。这两个产品都可免费用于商业和非商用目的，并且在Sun公开许可证（SPL）的限制下可获得供重用的源码。



22.1.1 NetBeans平台概述

NetBeans平台是一个应用程序框架，为开发大多数的桌面应用程序提供了必备的环境。NetBeans平台提供了大多数桌面应用程序所需要的一些功能，例如菜单、文档管理和设置等。在基于这个平台开发应用程序的时候，开发人员不必反复编写相同的代码。只需要编写要实现的模块，然后将这些模块与NetBeans平台捆绑在一起，这样就可以拥有一个外观漂亮且具有标记的跨平台应用程序。使用NetBeans平台构建桌面应用程序，可以节省大量的开发时间。此平台为几乎所有的大型桌面应用程序提供了通用的服务（窗口、菜单、设置管理和存储，以及文件访问等），因此，开发者可以完全将精力投入到各自的业务逻辑上。

NetBeans集成开发环境是一个开发环境，是一个供程序员编写、编译、调试和部署程序的工具。前面所有章节的介绍都是围绕着NetBeans集成开发环境来进行讲解的。NetBeans集成

Download at http://www.pin5i.com/

开发环境中包含了在 NetBeans 平台上构建应用程序所需的一切功能。

22.1.2 NetBeans 平台的功能

通过 NetBeans 平台，开发者可以集中精力开发应用程序的重要部分，即每个应用程序独有的业务逻辑，从而节省大量的时间和精力。下面是 NetBeans 平台所提供的一些功能。

- 用户界面管理：该平台提供了窗口、菜单、工具栏和其他表示组件。开发者只需编写由系统管理的操作和组件即可，这样不但节省了时间，而且生成的代码也更加清晰简捷、错误更少。
- 数据和表示管理：NetBeans 平台包含了一组丰富的工具，可为用户显示数据并处理数据。
- 设置管理：NetBeans 平台提供了保存和恢复设置，即使对于复杂的业务对象，也可以确保其安全、简单和透明，并且这些设置通常是自动完成的。
- 图形编辑：通过图形库，可以使用拖放的方式轻松地创建数据的图形化视图。
- 编辑器：作为该平台的一种扩展，在 NetBeans 上构建的应用程序可以使用 NetBeans 编辑器，它是一个可以构建定制编辑器的功能强大且可扩展的工具集。
- 向导框架：此工具集用于轻松构建可扩展的、用户友好的向导以指导用户完成更多复杂的任务。
- 存储管理：一种抽象的基于文件的数据访问方式。在 NetBeans 范例中，“文件”是指本地文件，或位于 FTP 服务器、CVS 库、XML 文件或数据库中的远程数据。对于使用该数据的其他模块来说，此数据的存储位置是完全透明的。
- 具有大量的附加模块选择：诸如版本控制支持、专用的编辑器、专用的 UI 组件、通过 FTP 和其他传输方式获得的远程数据访问，并且提供了使用各种 Java 和互联网技术的简便方法（可作为插件供任何应用程序使用）。
- 基于互联网的更新传送：基于 NetBeans 的应用程序可以使用 Java Web Start 技术，为复杂的应用程序提供一组基于用户角色的定制模块。支持 Web Start 的应用程序始终可以保持最新状态，并且将集中管理和部署与丰富的客户端用户体验优势集于一身。对于非 Web Start 应用程序，可以通过自动更新模块（可选组件）从 Web 站点上下载更新或新功能。

22.2 NetBeans 模块开发

NetBeans 模块是一个包含 Java 类的归档（JAR）文件，这些编写的类用于与 NetBeans API 进行交互。模块通过其 MANIFEST.MF 文件中的条目将自身标识为一个模块。

NetBeans 模块开发包括扩展 NetBeans 集成开发环境和构建功能丰富的独力的客户端应用程序。本节主要讲解如何扩展 NetBeans 集成开发环境。

22.2.1 创建 NetBeans 插件模块项目

在 NetBeans 集成开发环境中，选择“文件”→“新建项目”命令，打开“新建项目”对话框。在“类别”列表中，选择“NetBeans 模块”节点，如图 22-1 所示。

NetBeans 插件模块支持提供以下三种项目类型。

- 模块：为单独插件模块创建模板。
- 库包装模块：为一个或多个插件模块所需的外部 JAR 文件创建插件模块。
- 模块套件：用于创建可以独立运行的富客户端应用程序。

Download at http://www.pin5i.com/

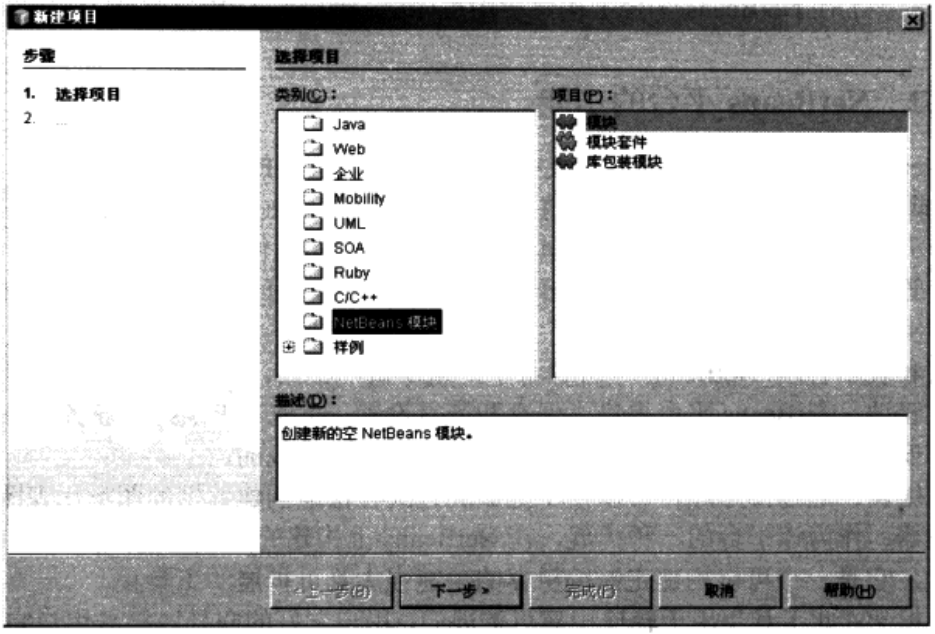


图 22-1 选择 NetBeans 插件模块项目

这里在“项目”列表中选择“模块”节点，单击“下一步”按钮，打开“新建 模块”对话框，如图 22-2 所示。

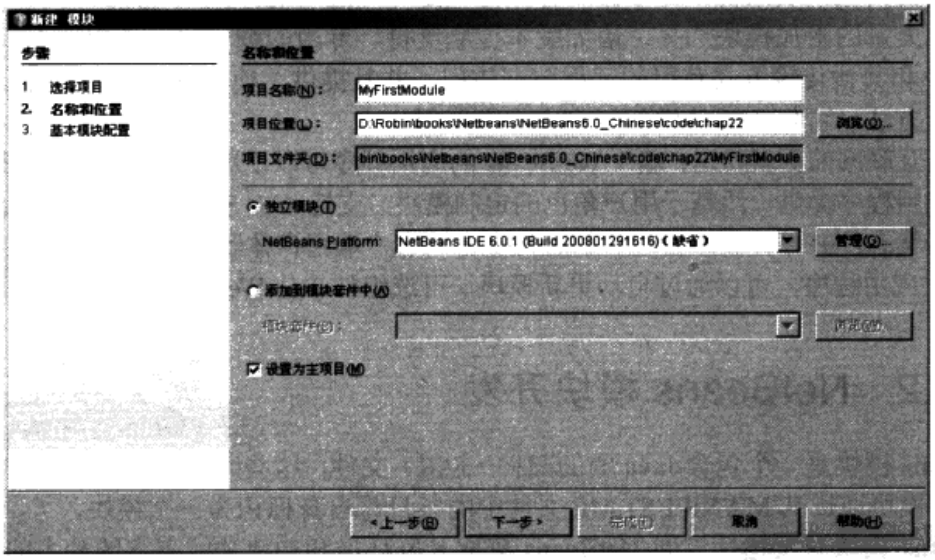


图 22-2 “新建模块”对话框

在“项目名称”文本框中，输入项目名称 MyFirstModule。在“项目位置”文本框中输入项目的位置，或者单击“浏览”按钮选择项目的位置。

如果要创建独立的模块，则要选中“独立模块”单选按钮，然后在“NetBeans Platform”下拉列表框中选择模块运行的 NetBeans 平台。如果要删除或者添加其他的平台，则可以单击“管理”按钮，打开如图 22-3 所示的“NetBeans 平台管理器”对话框。

如果要把模块添加到其他的模块套件中，则要选中“添加到模块套件中”单选按钮，然后在“模块套件”下拉列表框中选择模块套件。

选中“设置为主项目”复选框，单击“下一步”按钮，打开“基本模块配置”对话框，如图 22-4 所示。

Download at http://www.pin5i.com/

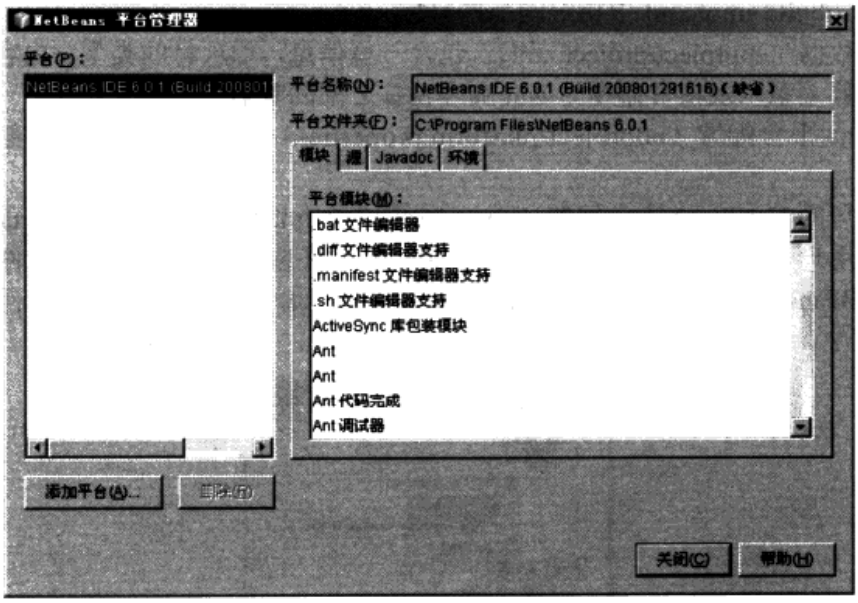


图 22-3 “NetBeans 平台管理器”对话框

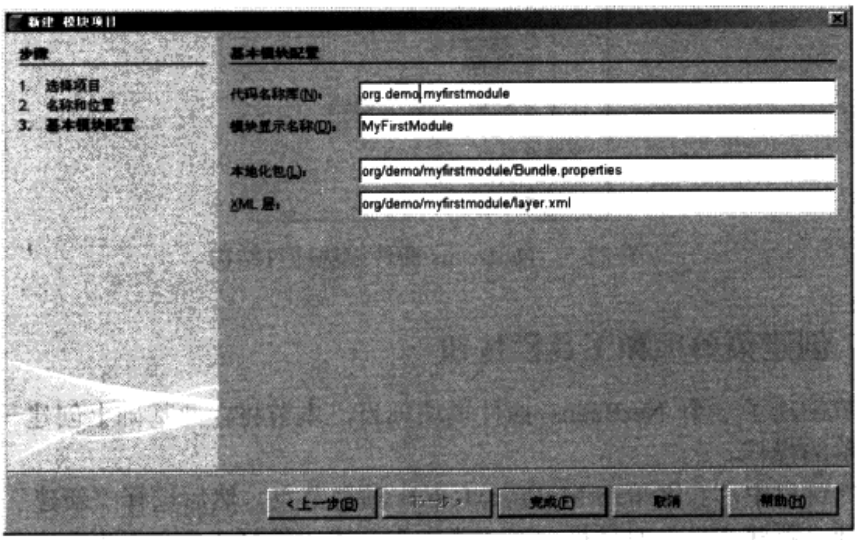


图 22-4 “基本模块配置”对话框

在“代码名称库”文本框中输入库名字，这里输入 org.demo.myfirstmodule。在“模块显示名称”文本框中输入模块的名字，这里输入 MyFirstModule。

“本地化包”文本框用于输入本地话的包名字和路径，该包用于为国际化指定特定于语言的字符串。“XML 层”文本框用于输入 layer.xml 文件的路径，该文件用于在 NetBeans 系统文件系统中注册菜单和工具栏按钮等项。本地化包和 XML 层将存储在 org.demo.myfirstmodule 包中。

单击“完成”按钮，将创建一个 NetBeans 插件模块项目。该项目包含所有源代码和项目元数据，如项目的 Ant 生成脚本。在“项目”窗口中展开项目，可以看到如图 22-5 所示的逻辑结构。除本地化包和 XML 层外，该项目还包含以下重要文件。

- 模块清单（manifest.mf）：声明此项目属于插件模块。此外，它还设定了一些特定于模块的设置，如 XML 层的位置、本地化包的位置以及模块的版本。
- 生成脚本（build.xml）：提供一个位置，可以在其中创建自己的 Ant 目标并覆盖

Download at http://www.pin5i.com/

nbproject/build-impl.xml 中指定的那些目标。

- 项目元数据（nbproject/project.xml）：包含一些信息，如项目的类型、内容、平台、类路径、依赖关系以及项目命令与 Ant 脚本目标之间的映射。
- NetBeans 平台配置（nbproject/platform.properties）：包含集成开发环境或平台使用的属性。
- 每个用户的 NetBeans 平台配置（nbproject/private/platform-private.properties）：包含特定于每个用户（当前用户）的 NetBeans 集成开发环境安装的属性。该文件中的设置优先于 NetBeans 平台配置文件中的设置。当使用 CVS 共享代码时，该文件不会被导入到 CVS 库中。

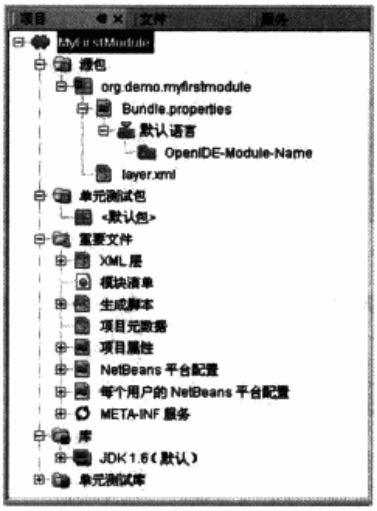


图 22-5 NetBeans 插件模块项目结构



22.2.2 创建菜单项和工具栏按钮

上一个小节创建了一个 NetBeans 插件模块项目，本节将在此基础上创建一个菜单命令，并添加一些简单的操作。

在“项目”窗口中，右键单击 MyFirstModule 项目节点，然后选择“新建”→“文件/文件夹”命令，打开“新建文件”对话框。在“类别”列表中选择“模块开发”节点，此时可以创建如图 22-6 所示的几种文件。

- Java SE 库描述符：创建一个 Java SE 库描述符文件，用于将新的类库添加到用户的库管理器中。可以包含关联的源代码和/或 Javadoc。
- 语言支持：为某些特定 MIME 类型和文件扩展名创建通用语言框架 nbs 文件。可以在 nbs 文件内定义对多种编辑功能（语法着色、缩排、代码折叠和导航等）的支持。
- 操作：创建可添加到菜单、上下文菜单和工具栏等中的 NetBeans 操作（CallableSystemAction 或 CookieAction）。
- JavaHelp 帮助集：使用此模板可以在模块中包括 JavaHelp。
- 文件类型：用于声明 NetBeans 将识别的文件类型。
- 模块安装程序：使用此模板可以添加一个用于管理模块安装以及卸载生命周期的类。
- 选项面板：使用此模板可以在此模块中包括一个新的选项面板。
- 项目模板：使用此模板可以将新的模板添加到“新建模板”向导中。这可用于捆绑使用特殊技术的样例应用程序，或者设置某种复杂的项目基础结构（如用于自由格式的

Download at http://www.pin5i.com/

项目）。

- 更新中心：使用此模板可以在该模块中注册一个更新中心。模块套件上下文菜单中的“创建 NBM...”将生成更新描述符（XML 文件）。此 XML 文件必须可以通过某个 URL 进行公开访问。当用户使用此模板安装模块时，自动更新向导将自动在指定的更新中心中进行查找。
- 窗口组件：创建新的单一窗口，它与窗口系统以及主菜单中打开它的操作集成在一起。
- 向导：创建一个新向导，可以将其注册为“新建文件”向导，或者从定制代码（例如，从表示全局的弹出式菜单的操作）中进行调用。

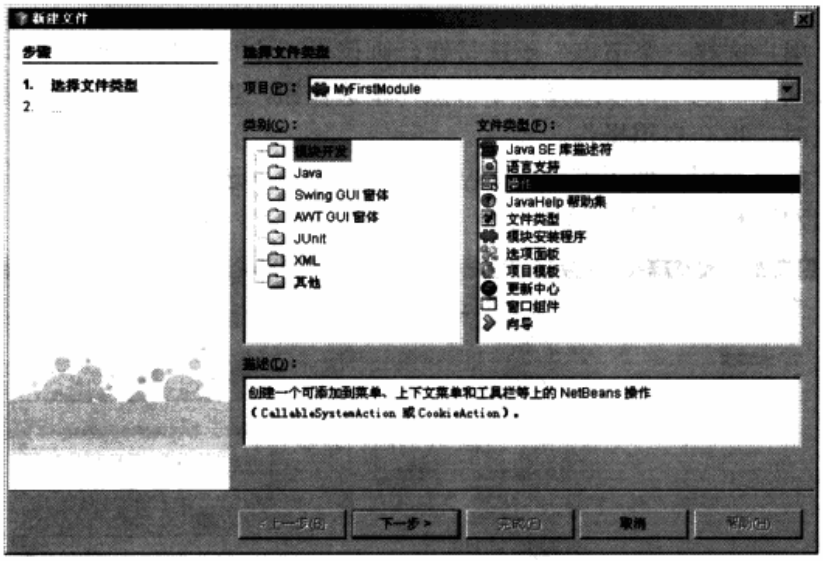


图 22-6 “新建文件”对话框

这里在“文件类型”列表中选择“操作”节点，创建一个菜单命令。单击“下一步”按钮，打开“操作类型”对话框，如图 22-7 所示。

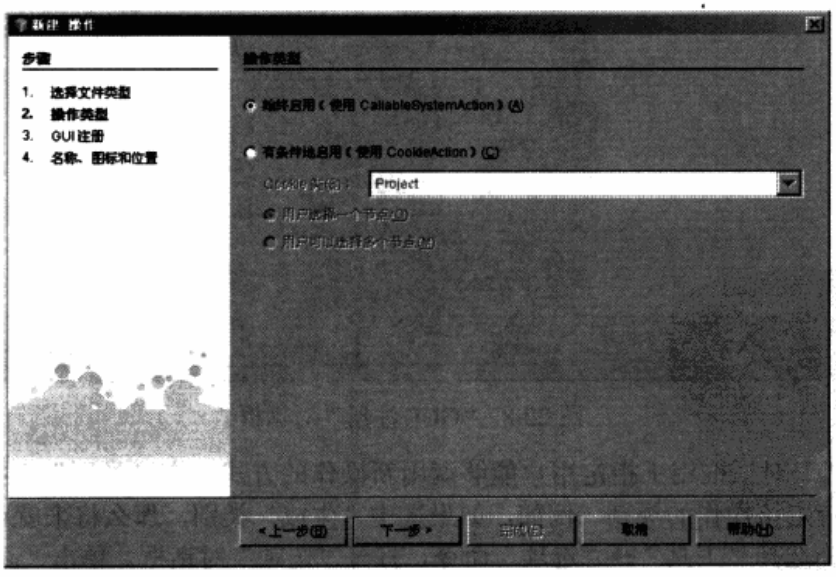


图 22-7 “操作类型”对话框

如果选中“始终启用（使用 CallableSystemAction）”单选按钮，则指定通过菜单栏中的菜单项、工具栏中的工具栏按钮或集成开发环境中任意位置处的快捷键调用操作。即该操作为始

Download at http://www.pin5i.com/

终启用的。

如果选中“有条件地启用（使用 CookieAction）”单选按钮，则在一定条件下才能启动该操作。在“Cookie 类”下拉列表框中选择使用的 Cookie 类，这会为对其派生子类的所有类创建操作：

- DataObject
- EditCookie
- Project
- EditorCookie
- OpenCookie

如果选中“用户选择一个节点”单选按钮，则仅当选定了为其创建操作的节点时，才会启用该操作；如果选中“用户可以选择多个节点”单选按钮，当选定多个节点而不是只选定为其创建操作的节点时，也会启用操作。

这里选中“始终启用（使用 CallableSystemAction）”单选按钮，单击“下一步”按钮，打开“GUI 注册”对话框，如图 22-8 所示。

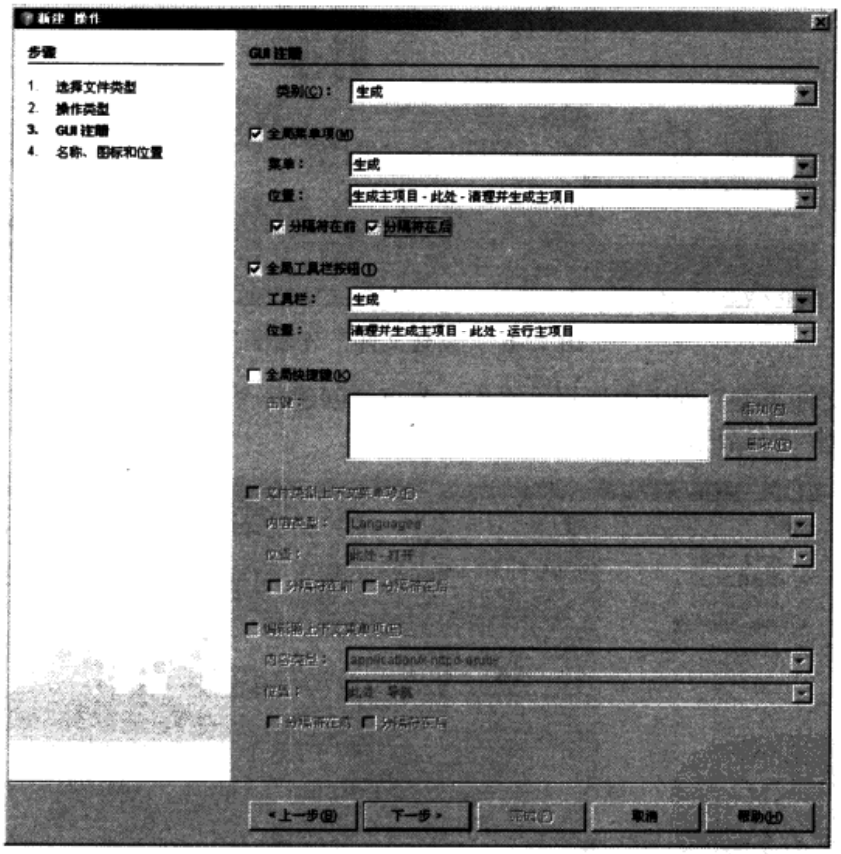


图 22-8 “GUI 注册”对话框

“GUI 注册”对话框用于指定用户能够调用新操作的方式。在“类别”下拉列表框中选择某一个类别，指定该操作的类别。例如，如果选中“生成”类别，那么将来就可以在 NetBeans 集成开发环境中选择“工具”→“选项”命令，打开“选项”对话框。单击“高级选项”按钮，切换到“高级选项”对话框。打开“IDE 配置”→“操作”节点，并在“生成”节点下找到创建的操作，如图 22-9 所示。

Download at http://www.pin5i.com/

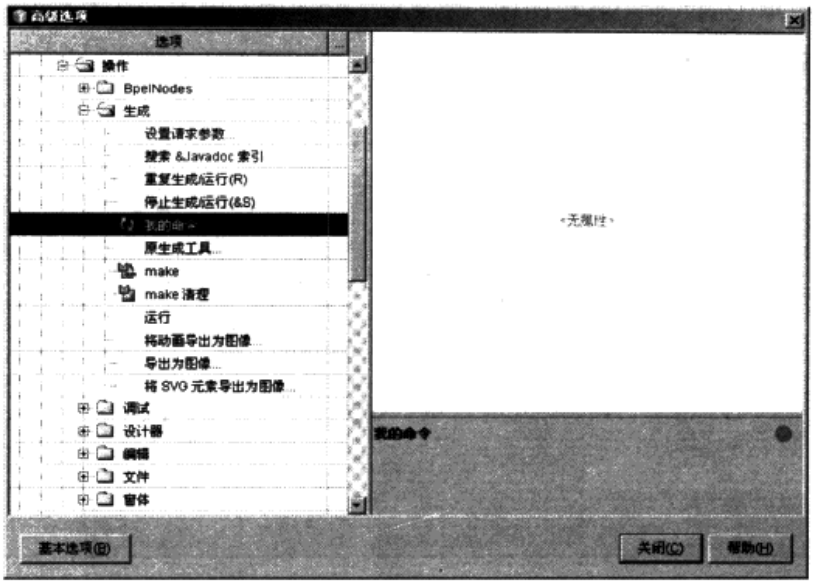


图 22-9 操作的类别

选中“全局菜单项”复选框，指定将操作显示为某个菜单的菜单项。在“菜单”下拉列表框中选择要出现在哪个菜单中。在“位置”下拉列表框中选择出现的具体位置，例如选择“Java DB 数据库-此处-组件库管理器”，那么表示新建的操作将位于“Java DB 数据库”命令和“组件库管理器”命令之间。

这里“菜单”下拉列表框中选择“生成”选项，然后在“位置”下拉列表框中选择“生成主项目-此处-清理并生成主项目”选项。选中“分隔符在前”复选框，指定是在其前面显示分隔符；选中“分隔符在后”复选框，指定在其后面显示分隔符。

选中“全局工具栏按钮”复选框，指定为操作创建工具栏按钮。在“工具栏”下拉列表框中选择按钮出现在哪个工具栏上，这里选择“生成”。在“位置”下拉列表框中选择按钮出现的位置，这里选择“清理并生成主项目-此处-运行主项目”，表示在“清理并生成主项目”按钮和“运行主项目”按钮之间创建该操作的按钮。

如果选中“全局快捷键”复选框，则可以为该操作指定全局的快捷键。单击“添加”按钮会打开“添加快捷键”对话框，如图 22-10 所示。按下任意的快捷键，会在“快捷键”文本框中自动记录下该快捷键。单击“确定”按钮，即可把输入的快捷键分配给操作。

仅当在如图 22-7 所示的“操作类型”对话框中选择了“有条件地启用（使用 CookieAction）”单选按钮时，才能设置以下项。

- 文件类型上下文菜单项：指定在弹出式菜单中显示操作的文件类型。
- 编辑器上下文菜单项：指定在弹出式菜单中显示操作的编辑器。

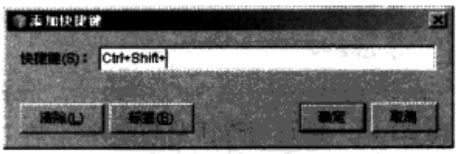


图 22-10 “添加快捷键”对话框

设置完成后，单击“下一步”按钮，打开“名称、图标和位置”对话框，如图 22-11 所示。

在“类名”文本框中指定新操作类的名称，这里输入 MyAction。在“显示名称”文本框中指定操作的标签，该标签将显示在界面上。这里输入“我的命令”。

如果在如图 22-7 所示的“GUI 注册”对话框中选择了选择了“全局菜单项”或者“全局工具栏按钮”复选框，则需要为操作准备一个 16×16 像素的图标和一个 24×24 像素的图标。其中 16×16 像素的图标显示在菜单命令旁边，以及作为小工具栏图标显示（在工具栏上单击鼠标

Download at http://www.pin5i.com/

右键，选择弹出菜单中的“小工具栏图标”命令）；24×24 像素的图标作为普通工具栏图标显示。

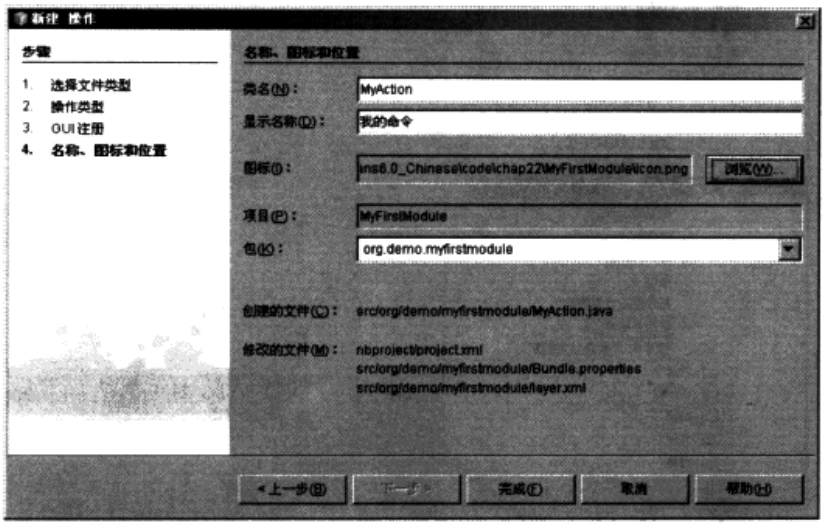


图 22-11 “名称、图标和位置”对话框

单击“浏览”按钮，查找文件系统中的某个 16×16 像素图标。请确保在相同目录下存在一 24×24 像素的图标，该图标具有相同的名称，但名称附加了“24”。例如 16×16 像素图标名字为 icon.png，那么 24×24 像素图标名字应该为 icon24.png。



说明：如果在如图 22-8 所示的“GUI 注册”对话框中只选择了“全局菜单项”复选框，那么可以不用指定图标（即菜单命令没有图标），此时只会出现一个警告；如果选中了“全局工具栏按钮”复选框，则必须指定图标，否则无法创建操作。

设置完毕后，单击“完成”按钮，即可创建一个操作，此时的“项目”窗口中会显示创建的操作及其图标文件，如图 22-12 所示。



图 22-12 创建的操作及其图标



22.2.3 安装和卸载插件模块

以上创建了一个简单的 NetBeans 插件模块项目，并创建了一个空白的 NetBeans 菜单命令和工具栏按钮。此时，选中该命令不会执行任何的操作，在第 22.4 节中将介绍如何为操作编写执行代码。

在“项目”窗口中，右键单击 NetBeans 插件模块项目节点，选择弹出菜单中的“在目标平台中安装/重新装入”命令，将会在目标平台中安装插件。此时，会在打开的 NetBeans 平台中看到创建的操作，如图 22-13 所示。

在工具栏中可以看到创建的工具栏按钮，如图 22-14 所示。在工具栏上单击鼠标右键，选择弹出菜单中的“小工具栏图标”命令，则会显示如图 22-15 所示的图标。

选择“工具”→“插件”命令，打开“插件”对话框，打开“已安装”标签页，如图 22-16 所示。选中“MyFirstModule”节点，单击“卸载”按钮，可以卸载已经安装的模块。

在“项目”窗口中，右键单击 NetBeans 插件模块项目节点，选择弹出菜单中的“在开发 IDE 中安装/重新装入”命令，将会在当前 IDE 中安装插件。这里不推荐使用该命令，因为一旦你的模块出现问题就会可能导致 NetBeans 集成开发环境的运行出现问题。

Download at http://www.pin5i.com/

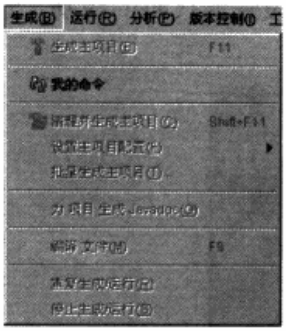


图 22-13 创建的菜单命令



图 22-14 大工具栏图标



图 22-15 小工具栏图标

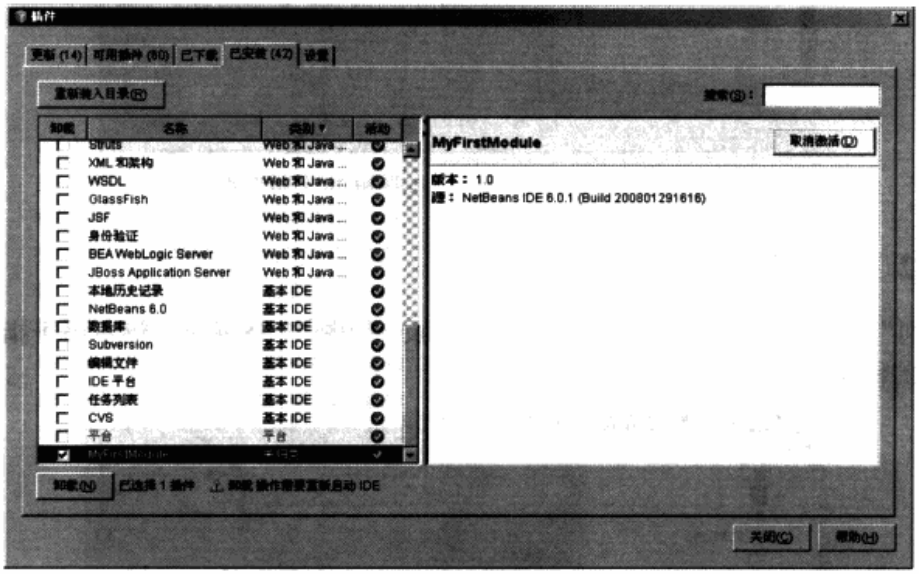


图 22-16 在“插件”对话框中卸载模块

在“项目”窗口中，右键单击 NetBeans 插件模块项目节点，选择弹出菜单中的“创建 NBM”命令，将会生成 NBM 文件。NBM 文件是为通过 Web 进行传送而打包的 NetBeans 模块。NBM 文件和模块 JAR 文件之间的主要区别是：

- NBM 文件可以包含多个 JAR 文件，模块可以将其使用的任何库打包到其 NBM 文件中；
- NBM 文件包含元数据，NetBeans 将使用它在更新中心中显示其相关信息，如清单内容、许可证等；
- NBM 文件只是具有特殊扩展名的 ZIP 文件。它们使用 JDK 的机制对 JAR 文件进行签名。一般情况下，无须担心 NBM 文件内容的问题，只需让用于 NBM 创建的标准 Ant 生成脚本进行处理即可。

22.3 插件模块项目管理

在创建了 NetBeans 插件模块项目后，可以在“项目”窗口中右键单击项目，选择弹出菜单中的“属性”命令，打开“项目属性”对话框。通过该对话框可以对项目进行管理。

22.3.1 源和代码级别

在“类别”列表中选择“源”节点，打开“源”页，如图 22-17 所示。在该页的“项目文

Download at http://www.pin5i.com/

件夹”文本框中显示了项目文件夹。“源代码级别”下拉列表框用于设置源代码的级别，默认值是 1.4。

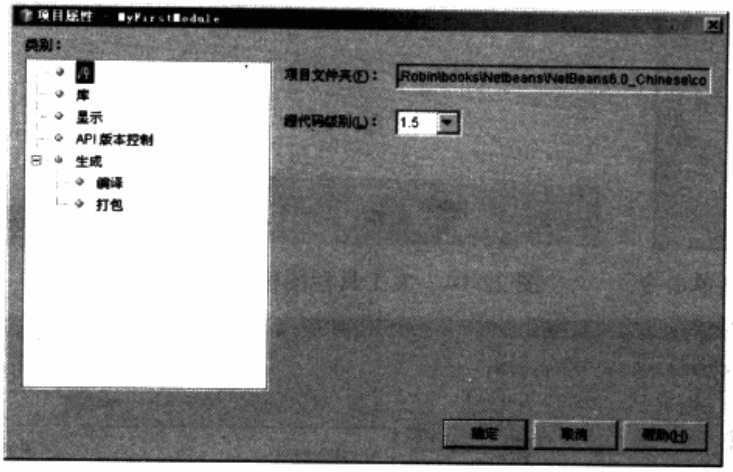


图 22-17 “项目属性”对话框的“源”页

22.3.2 库管理

在“类别”列表中选择“库”节点，打开“库”页，如图 22-18 所示。该页用于指定 NetBeans 平台、模块的依赖关系以及所需的标记。

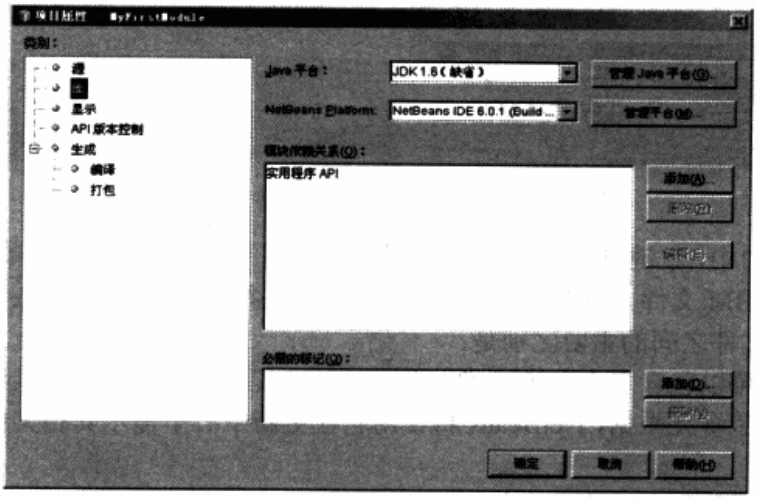


图 22-18 “项目属性”对话框的“库”页

“Java 平台”下拉列表框指定用于生成模块项目或模块套件项目的目标 JDK。单击“管理 Java 平台”按钮，可以打开 Java 平台管理器添加或者删除 Java 平台。

“NetBeans 平台”下拉列表框用于指定编译和运行当前模块的目标 NetBeans 平台。可用的 NetBeans 平台是指那些已在 NetBeans 平台管理器中注册的平台。单击“管理平台”按钮可以打开 NetBeans 平台管理器，添加或者删除 NetBeans 平台。

在“模块依赖关系”列表中显示了按字母顺序排列的模块依赖关系列表，每个列表条目将显示为模块的本地化显示名称。



说明：在“模块依赖关系”列表中添加的 API，会保存在 nbproject/project.xml 文件中。

Download at http://www.pin5i.com/

单击“添加”按钮，可以打开“添加模块依赖关系”对话框，添加新的模块依赖关系，如图 22-19 所示。在“模块”列表中列出了可以使用的库。单击某个库，会在下面的“描述”文本框中显示该库模块的描述信息。用户可以直接在“模块”列表中选择要添加的库，然后单击“确定”按钮进行添加。如果在“过滤器”文本框中输入某个过滤关键字，则模块依赖关系列表将会缩小范围，仅列出那些可以提供所键入的类的模块，如图 22-20 所示。

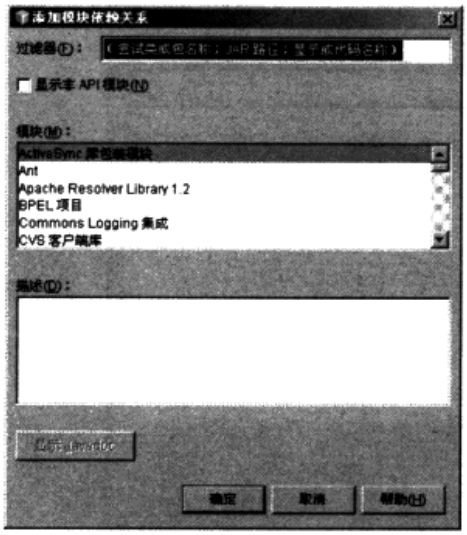


图 22-19 “添加模块依赖关系”对话框

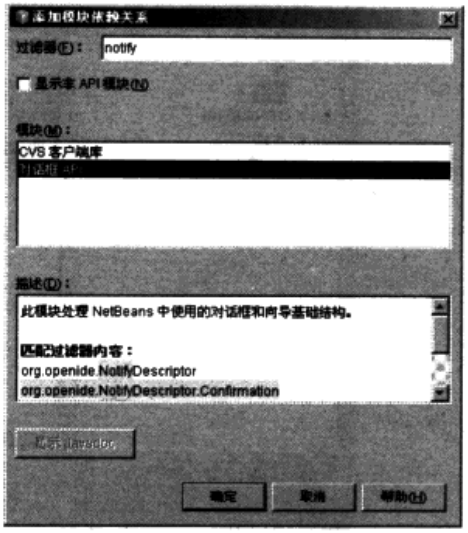


图 22-20 对模块进行过滤

如果选中“显示非 API 模块”复选框将显示所有的模块。如果取消选中状态，那么在“模块”列表中仅显示公共包中的模块，或者友元包中的友元模块。

找到所需的模块后，单击“确定”按钮，选中的模块将添加到如图 22-18 所示的“模块依赖关系”列表中，同时 NetBeans 集成开发环境将在 project.xml 文件中添加关联的条目。

如果在图 22-18 所示的“模块依赖关系”列表中选中某个模块，单击“删除”按钮，则会删除当前的依赖关系。如果单击“编辑”按钮，则会打开“编辑模块依赖关系”对话框，如图 22-21 所示。该对话框可以用来查看模块的 JAR 文件路径、修改版本号、包含在类路径中的 API 包等。

“库”页中的“必需的标记”列表显示了必须使用到的标记。单击“添加”按钮，可以打开“提供的标记：”对话框，选择添加标记，如图 22-22 所示。

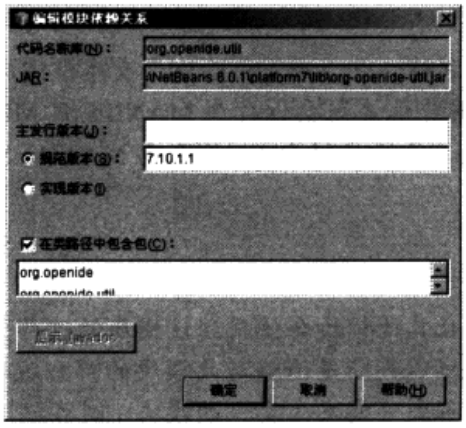


图 22-21 “编辑模块依赖关系”对话框

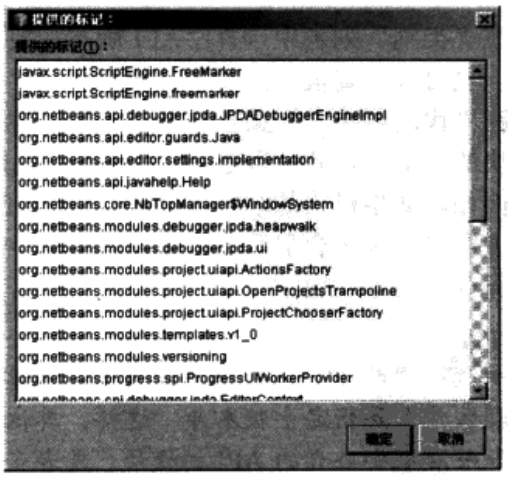


图 22-22 “提供的标记：”对话框

Download at http://www.pin5i.com/

22.3.3 设置显示参数

在“类别”列表中选择“显示”节点，打开“显示”页，如图 22-23 所示。该页可以添加将写入本地化包（Bundle.properties 文件）中的信息。

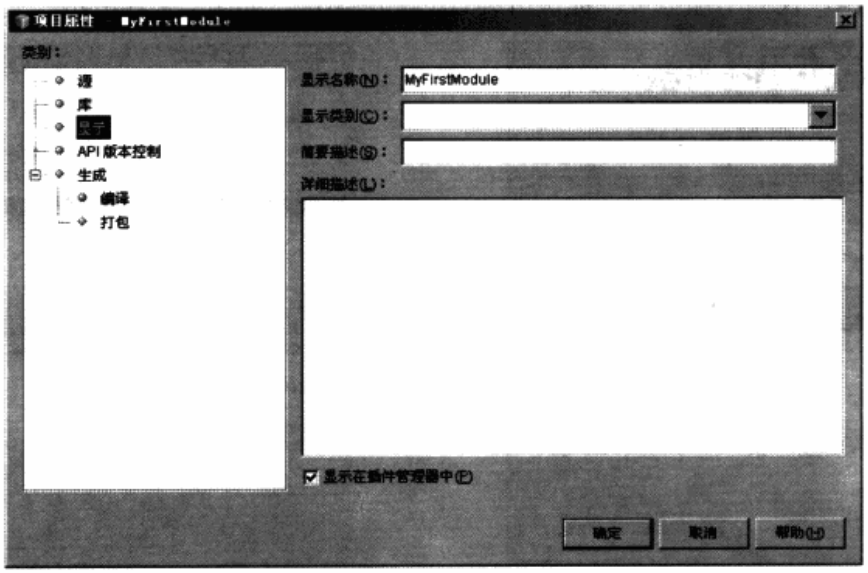


图 22-23 “项目属性”对话框的“显示”页

在“显示名称”文本框中设置模块的显示名称，该名称可以用于安装或者卸载模块，例如该名称将显示在如图 22-16 所示的插件管理器中。显示名称将在本地化包中设置 OpenIDE-Module-Name 值。

在“显示类别”下拉列表框中选择模块的类别。显示类别决定了在插件管理器中显示模块的位置。显示类别在本地化包中设置 OpenIDE-Module-Display-Category 值。

“简要描述”文本框用于设置模块的简要描述信息，即在本地化包中设置 OpenIDE-Module-Short-Descriptio 的值。“详细描述”文本框用于设置模块的详细描述信息，即在本地化包中设置 OpenIDE-Module-Short-Description 的值。

22.3.4 版本控制

在“类别”列表中选择“API 版本控制”节点，打开“API 版本控制”页，如图 22-24 所示。该页可以设置模块的版本信息。

在“代码名称库”文本框中显示唯一标识模块的名称。在“主发行版本”文本框中可以输入模块的主版本，可以为空或非负整数。代码库名称和主版本都会保存在 MANIFEST.MF 文件中。例如，如果模块的名字为 org.demo.myfirstmodule，主版本号为 1，那么在 MANIFEST.MF 文件中主发行版本会显示为一个正斜杠和一个数字：

```
org.myorg.module/1
```

说明：如果在模块中公开了某些 API，并且其他模块会使用到该模块的这些 API，那么就需要指定主版本号。这样，其他模块中就可以判断是依赖与此模块的哪个版本了。如果没有要公开的 API 的模块，或者变化很小或不太可能发生变化的模块，则可以不指定主发行版本。

Download at http://www.pin5i.com/

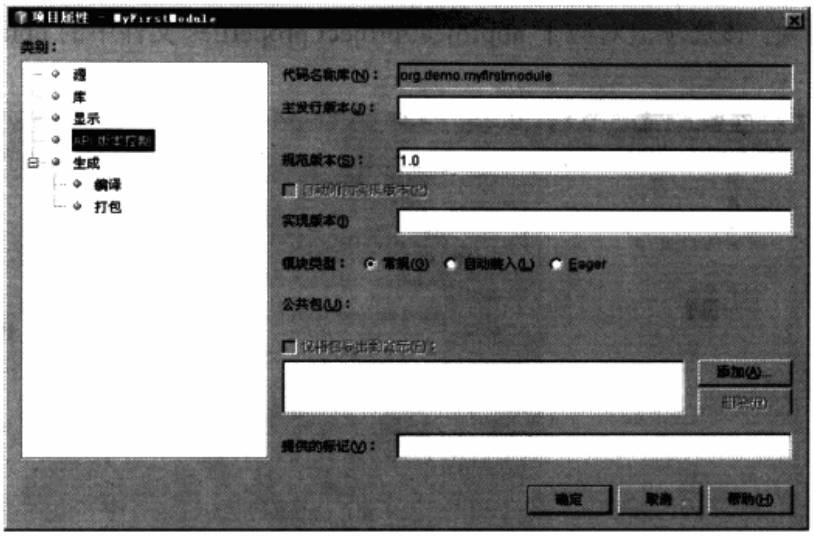


图 22-24 “项目属性”对话框的“API 版本控制”页

“规范版本”文本框用于指定规范的版本号，该版本号会在添加公开发布的方法时，自动递增 0.1，即 1.0、1.1、1.2，依此类推。“规范版本”文本框设置的值会保存为 MANIFEST.MF 文件中的 OpenIDE-Module-Specification-Version 属性值。

选中“自动附加实现版本”复选框，则会自动附件实现版本。该复选框只有在下面的“实现版本”文本框中输入任何版本后才能被激活。实现版本的值对应于 MANIFEST.MF 文件中的 OpenIDE-Module-Implementation-Version 属性值。

模块类型用于设置模块的装入类型，可以有如下的几个选项。

- 常规：该类型的模块将在平台启动被自动装入。
- 自动装入：仅当一个模块需要另一个模块时才装入的模块。例如，仅当依赖于某个库包装模块的模块需要它时，才装入该模块。
- Eager：如果满足模块的所有依赖关系，则会装入该模块。例如，为 NetBeans 文件提供着色服务的模块，依赖于编辑器模块。在禁用编辑器模块时，NetBeans 将自动禁用着色模块，否则则会自动装入该模块。

“公共包”列表中显示了源目录中包含的所有包以及任何 class-path-extensions 的列表。每一个列表选项旁边都有一个复选框，如果选中该复选框，则会在 nbproject/project.xml 文件的 public-packages 部分包含该包。

“仅将包导出到友元”列表用于显示可能包含其他模块的代码名称库，并且按排序进行排序。该列表只有在至少有一个包含的条目处于选中状态时，才为激活状态。此列表如果不为空，则列表中的包会保存在 friend-packages 中，并且在 nbproject/project.xml 中列出友元。

“提供的标记”文本框用于设置模块的标记，该值对应于 MANIFEST.MF 文件中的 OpenIDE-Module-Provides 属性值。

22.3.5 打包设置

在“类别”列表中选择“打包”节点，打开“打包”页，如图 22-25 所示。该页可以指定从项目中生成 NBM 文件的相关数据。

“JAR 文件”文本框用于显示将为模块生成的实际 JAR 文件的位置和名称，格式为 ${cluster}/${module.jar}。

选中“安装时需要重新启动”复选框，则安装该模块后必须重新启动 NetBeans 集成开发

Download at http://www.pin5i.com/

环境才能启用模块。该选项值对应于 nbproject/project.properties 文件中的 nbm.needs.restart 属性值。

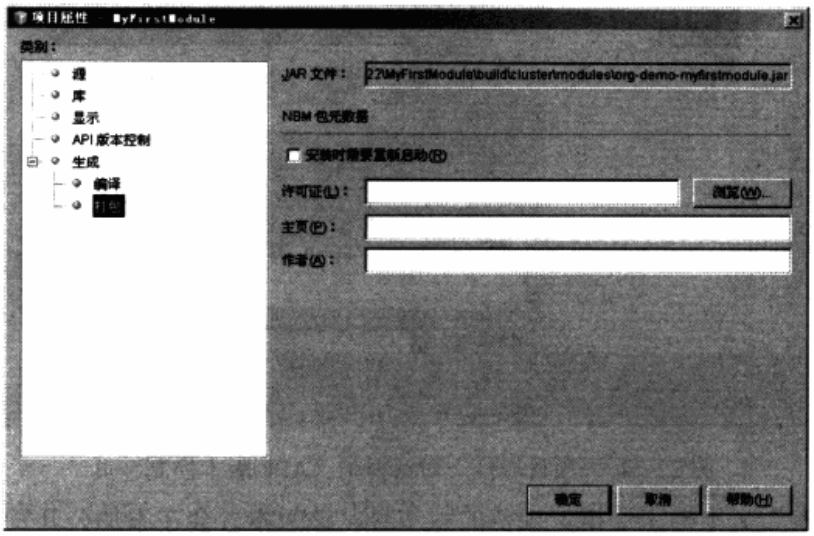


图 22-25 “项目属性”对话框的“打包”页

“许可证”文本框用于设置安装模块时，显示的指向 NBM 文件的许可证文件（可以是绝对路径或项目的相对路径）。单击“浏览”按钮可以在文件系统中查找许可证文件。该文本框中的值对应于 nbproject/project.properties 文件中的 license.file 属性值。

“主页”文本框用于设置模块相关的主页 URL，该值对应于 nbproject/project.properties 文件中的 nbm.homepage 属性值。

“作者”文本框用于输入模块的作者，该值对应于 nbproject/project.properties 文件中的 nbm.module.author 属性值。

22.4 为操作编写处理代码

在第 22.2 节中创建了一个空白的操作，本节将演示如何为操作编写简单的代码。在编写模块代码前，首先要确定这些代码中要使用到哪些 NetBeans 的 API，然后把它们添加到依赖的模块库中。

在下面的代码中要使用到 NetBeans 的对话框 API，因此添加过程介绍如下。

❶ 在 NetBeans 集成开发环境的“项目”窗口中右键单击 MyFirstModule 项目，选择弹出菜单中的“属性”命令，打开“项目属性”对话框。

❷ 在“类别”列表中选择“库”节点，打开“库”页。单击“模块依赖关系”列表旁边的“添加”按钮，打开“添加模块依赖关系”对话框。

❸ 在“过滤器”文本框中输入过滤关键字 notify，过一会后会显示如图 22-26 所示的模块。在“模块”列表中选择“对话框 API”选项，单击“确定”按钮，添加对对话框 API 的依赖关系。

单击“确定”按钮，返回 NetBeans 集成开发环境，至此添加了一个新的依赖关系。下面就可以进行编码了。

在 NetBeans 集成开发环境中打开 MyAction.java 文件，可以看到创建的 MyAction.java 文件内容如下所示：

Download at http://www.pin5i.com/

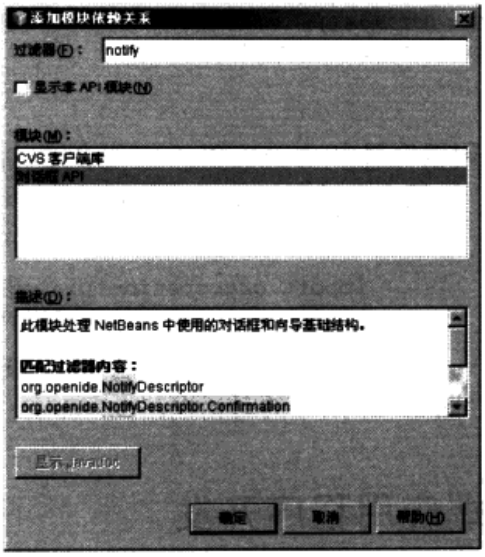


图 22-26 “添加模块依赖关系”对话框

```
package org.demo.myfirstmodule;

import org.openide.util.HelpCtx;
import org.openide.util.NbBundle;
import org.openide.util.actions.CallableSystemAction;

public final class MyAction extends CallableSystemAction {

    public void performAction() {
    // TODO implement action body
    }

    public String getName() {
        return NbBundle.getMessage(MyAction.class, "CTL_MyAction");
    }

    @Override
    protected String iconResource() {
        return "org/demo/myfirstmodule/icon.png";
    }

    public HelpCtx getHelpCtx() {
        return HelpCtx.DEFAULT_HELP;
    }

    @Override
    protected boolean asynchronous() {
        return false;
    }
}
```

其中，performAction()方法为操作的处理方法，选中操作后的响应代码应该添加到该方法中；getName()方法返回操作的名字；iconResource()方法返回操作的图标；getHelpCtx()方法返回该操作对应的帮助上下文。

这里，只为操作添加处理代码，因此在 performAction()方法中添加如下的代码：

```
public void performAction() {
    String msg = "我被执行了！";
    try {
        NotifyDescriptor d = new NotifyDescriptor.Message(
                msg, NotifyDescriptor.INFORMATION_MESSAGE);
        DialogDisplayer.getDefault().notify(d);
```

Download at http://www.pin5i.com/


```
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println(e.toString());
        }
    }
```

在 MyAction.java 的源代码编辑器中，按 Alt+Shift+F 组合键以允许 NetBeans 自动添加导入。这里会自动添加 DialogDisplayer 和 NotifyDescriptor 的导入，即添加如下的两行代码：

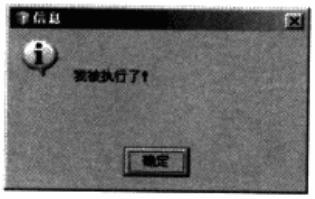


图 22-27 为操作添加的响应

```
import org.openide.DialogDisplayer;
import org.openide.NotifyDescriptor;
```

此时，再次在目标平台中运行项目，单击该操作后会显示如图 22-27 所示的对话框。

22.5 富客户端应用程序开发

在第 22.2 节和第 22.4 节介绍了如何开发一个 NetBeans 插件模块，本节将介绍如何开发一个基于 NetBeans 平台的独立的应用程序，即富客户端应用程序开发。

22.5.1 富客户端应用程序开发概述

NetBeans 提供了 NetBeans 平台，该平台是在其上构建应用程序的框架。实际上，NetBeans 集成开发环境就是基于这个平台的一个独立的应用程序，只不过这个应用程序还可以用来开发其他的应用程序。

NetBeans 平台提供了现成的应用程序常见功能，如菜单、文档管理和设置等。基于 NetBeans 平台开发富客户端应用程序时，只需要提供 NetBeans 平台尚不包含的应用程序部分，而不是从头开始编写应用程序。在开发周期结束时，可以将应用程序与 NetBeans 平台捆绑在一起，从而节省了很多时间和精力，并且构建的应用程序稳定可靠。

基于 NetBeans 平台的富客户端应用程序的每个不同部分是由单独的模块提供的，其中一些模块通过 Swing 库来提供用户界面元素。例如，如果富客户端应用程序是一个编辑器，则该应用程序中可能包含如下的两个模块：

- 提供语法突出显示的模块
- 提供文件模板的模块

下面是富客户端应用程序从创建到分发的开发过程。

❶ 通过使用模块套件项目模板来开始创建模块套件项目。

❷ 创建并开发插件模块项目（提供 NetBeans 平台不能提供的功能），该模块项目可以是一个或者多个。这里是主要的开发步骤。

❸ 如果模块项目是单独开发的，则要在模块套件项目中添加模块项目。

❹ 其他应用程序的设置（可选），例如去除不必要的模块、设置闪屏等。

❺ 生成并测试应用程序。

❻ 可以分发应用程序：创建 ZIP 分发或者创建 JNLP 应用程序。

❼ 分发应用程序的更新

22.5.2 创建模块套件项目

在 NetBeans 集成开发环境中，选择“文件”→“新建项目”命令，打开“新建项目”对

Download at http://www.pin5i.com/

话框。在“类别”列表中，选择“NetBeans 模块”节点。

在“项目”列表中选择“模块套件”节点，单击“下一步”按钮，打开“新建 模块套件”对话框，如图 22-28 所示。

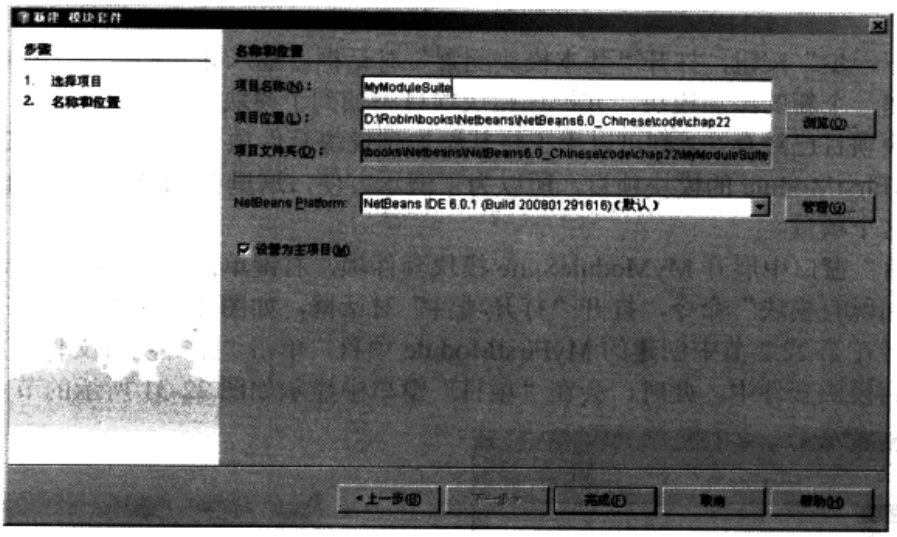


图 22-28 “新建 模块套件”对话框

在“项目名称”文本框中输入套件项目的名称 MyModuleSuite。在“项目位置”文本框中输入或者单击“浏览”按钮选择项目所在的目录。

在“NetBeans 平台”下拉列表框中选择开发的目标平台，这里使用默认平台即可。选中“设置为主项目”复选框，单击“完成”按钮即可创建一个空白的模块套件项目。



22.5.3 添加模块项目

在创建了模块套件项目后，需要创建或者添加模块项目，提供 NetBeans 平台所没有的功能。

在 NetBeans 集成开发环境中，选择“文件”→“新建项目”命令，打开“新建项目”对话框。在“类别”列表中，选择“NetBeans 模块”节点。在“项目”列表中选择“模块”节点，单击“下一步”按钮，打开“新建 模块”对话框，如图 22-29 所示。

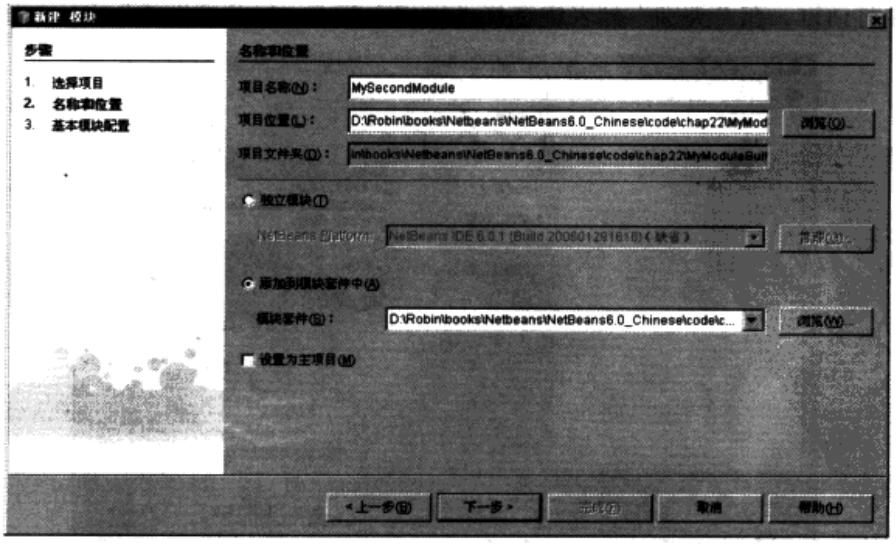


图 22-29 “新建 项目模块”对话框

Download at http://www.pin5i.com/

在“项目名称”文本框中输入模块项目的名称，例如 MySecondModule。在“项目位置”文本框中输入或者单击“浏览”按钮选择项目所在的目录。选中“添加到模块套件中”单选按钮，然后在“模块套件”下拉列表框中选择（或者单击“浏览”按钮选择）要添加到哪个模块套件项目中。

单击“下一步”按钮，打开“基本模块配置”对话框。输入必要的信息，单击“完成”按钮，即可创建一个新的添加模块，并把该模块项目添加到了模块套件项目中。

如果模块项目已经存在，则可以手工将其添加到模块套件项目中。因为在第 22.2 节已经创建了一个 MyFirstModule 的模块项目，所以为了演示方便，这里在 MyModuleSuite 模块套件项目手工添加这个项目。

在“项目”窗口中展开 MyModuleSuite 模块套件项，右键单击“模块”节点，选择弹出菜单中的“添加现有模块”命令，打开“打开项目”对话框，如图 22-30 所示。

这里选择在第 22.2 节中创建的 MyFirstModule 项目，单击“打开项目文件夹”按钮，将模块项目添加到模块套件中。此时，会在“项目”窗口中显示如图 22-31 所示的节点。

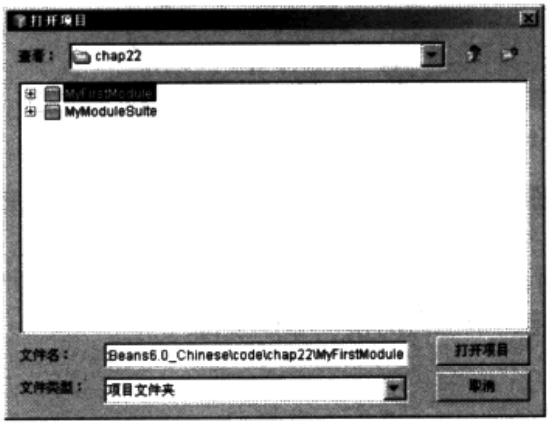


图 22-30 “打开项目”对话框



图 22-31 已经添加的模块项目



22.5.4 生成选项

在“项目”窗口中，右键单击模块套件项目，选择弹出菜单中的“属性”命令，打开模块套件项目的属性对话框。在“类别”列表中选择“生成”节点，打开“生成”页，如图 22-32 所示。

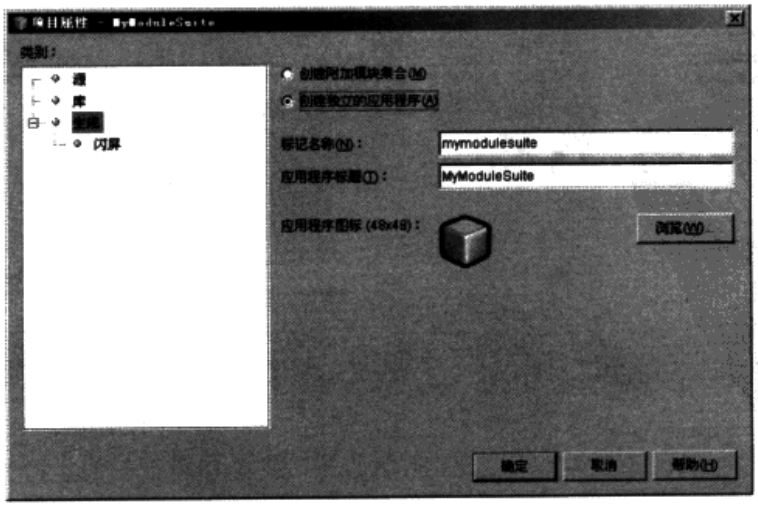


图 22-32 模块套件项目属性对话框的“生成”页

Download at http://www.pin5i.com/

选中“创建附加模块集合”单选按钮，将创建所有的附加模块。选中“创建独立的应用程序”单选按钮，将创建一个独立的应用程序。选中该单选按钮后，会显示如图 22-33 所示的“排除 IDE 模块”对话框，询问是否排除与 IDE 有关的模块。单击“排除”按钮删除多余的模块。图 22-34 是未排除 IDE 模块的“库”页面，图 22-35 是排除 IDE 模块后的“库”页面。

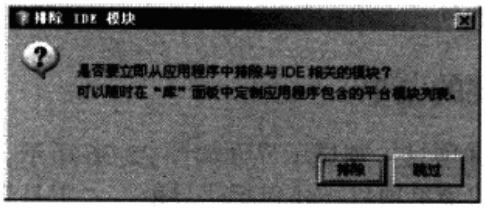


图 22-33 “排除 IDE 模块”对话框

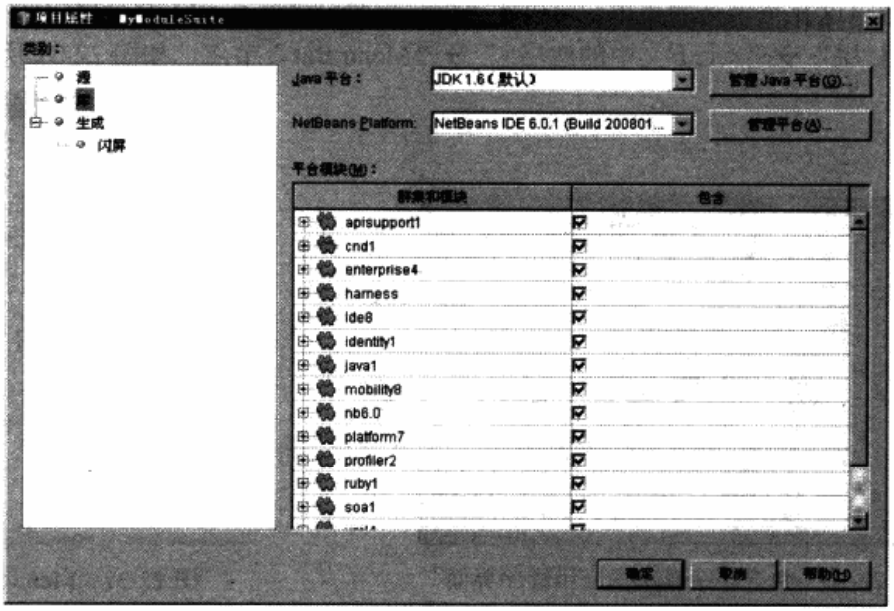


图 22-34 未排除 IDE 模块的“库”页面

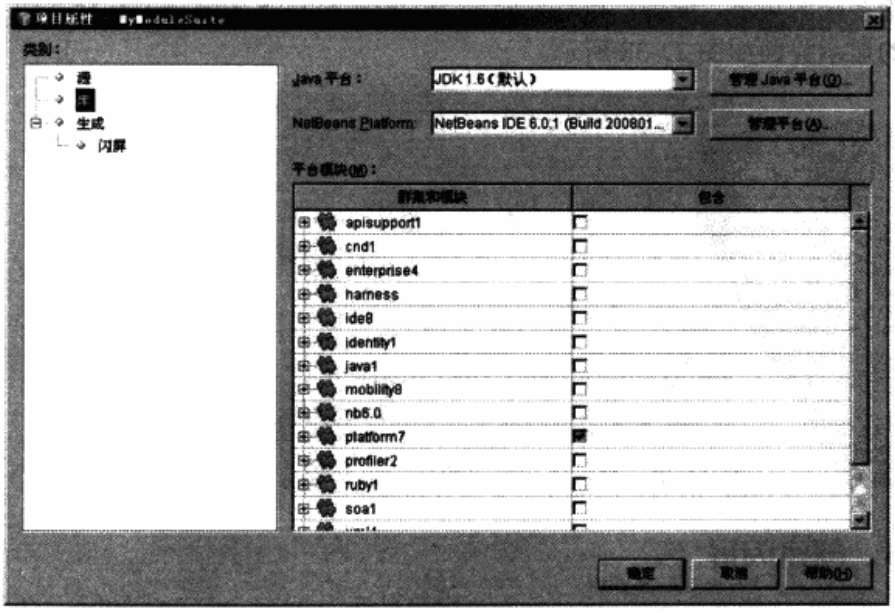


图 22-35 排除 IDE 模块后的“库”页面

Download at http://www.pin5i.com/

在“标记名称”文本框中输入应用程序可执行文件的名称。在“应用程序标题”文本框中输入应用程序标题栏中显示的标题。

单击“浏览”按钮可以选择应用程序“关于”对话框的图标，图标大小为 48×48 像素。该图标将显示在应用程序的“关于”对话框中，通过“帮助”菜单可访问该对话框。请注意，此图标不会设置可执行文件的图标。

22.5.5 去除不必要的菜单

此时，如果运行模块套件项目，则显示界面如图 22-36 所示。可以看到这里不仅包括了在 MyFirstModule 项目中创建的菜单项目，还包括“导航”、“工具”、“视图”和“窗口”等几个菜单。

如果要删除这几个多余的菜单，可以在“项目”窗口中展开“MyFirstModule”→“重要文件”→“XML 层”→“<上下文中的此层>”→“Menu Bar”节点，如图 22-37 所示。

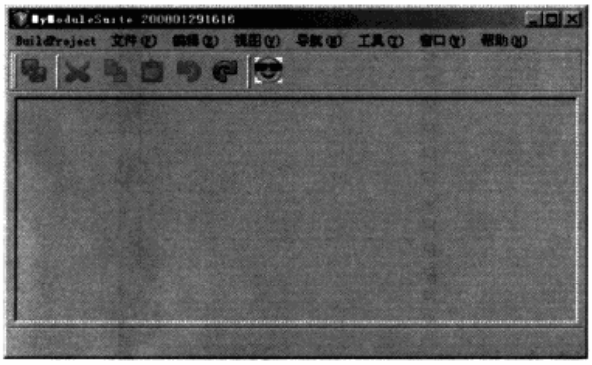


图 22-36 包含多余菜单的应用程序界面



图 22-37 Menu Bar 节点

这里可以把 Navigate、Tools、View、Window 等几个节点删除，这样就可以在富客户端应用程序中删除多余的菜单。同样，也需要删除 Toolbars 节点下的多余工具栏按钮。此时富客户端应用程序的显示界面如图 22-38 所示。

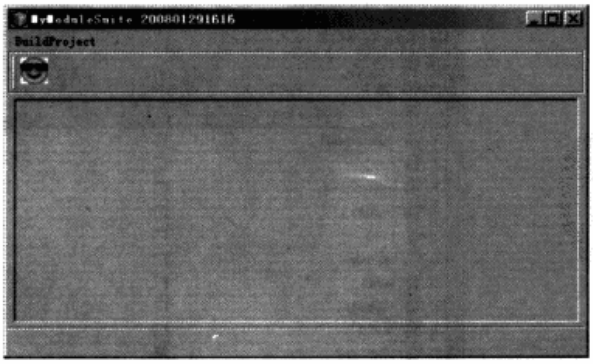


图 22-38 删除多余菜单后的富客户端应用程序界面

22.5.6 创建闪屏

图 22-39 是富客户端应用程序默认的闪屏，本节主要讨论如何在模块套件项目中，不通过编程就更改程序的闪屏。

Download at http://www.pin5i.com/

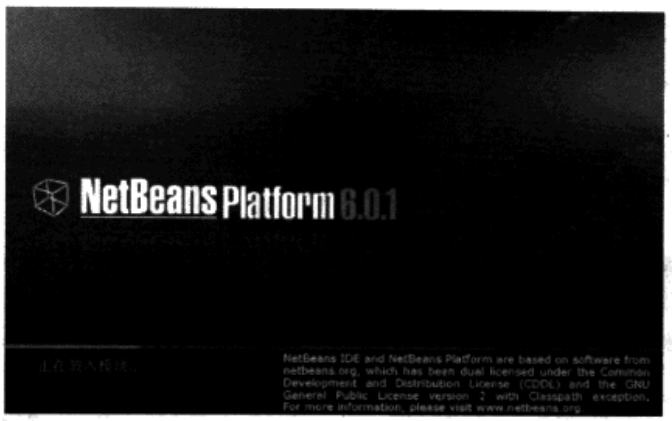


图 22-39 富客户端应用程序默认的闪屏

在“项目”窗口中，右键单击模块套件项目，选择弹出菜单中的“属性”命令，打开模块套件“项目属性”对话框。在“类别”列表中选择“生成”→“闪屏”节点，打开“闪屏”页，如图 22-40 所示。

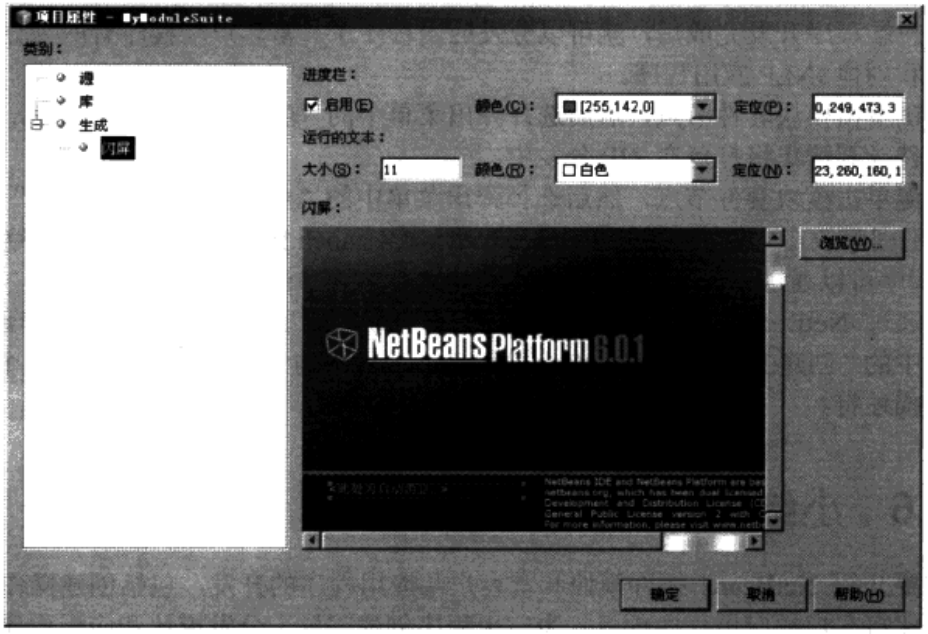


图 22-40 模块套件项目属性对话框的“闪屏”页

在该页中可以设置闪屏窗口中进度栏的如下属性。

- 启用：指定启动期间将显示的进度栏。
- 颜色：指定进度栏的颜色。
- 定位：指定进度栏的位置（以像素为单位）。

在“闪屏”页还可以设置在闪屏中显示的启动文本的属性：

- 大小：指定启动文本的大小。
- 颜色：指定启动文本的颜色。
- 定位：指定启动文本的位置（以像素为单位）。

单击“浏览”按钮，可以为闪屏窗口选择一副背景图像。对于如图 22-41 所示的设置，闪屏的显示界面如图 22-42 所示。

Download at http://www.pin5i.com/

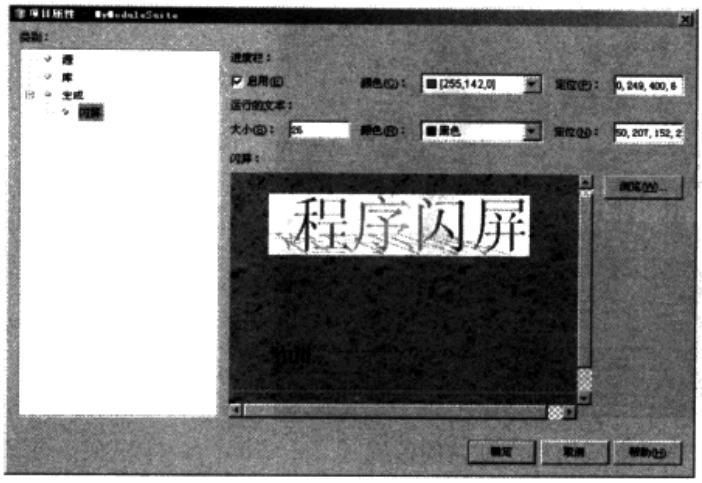


图 22-41 设置新的闪屏

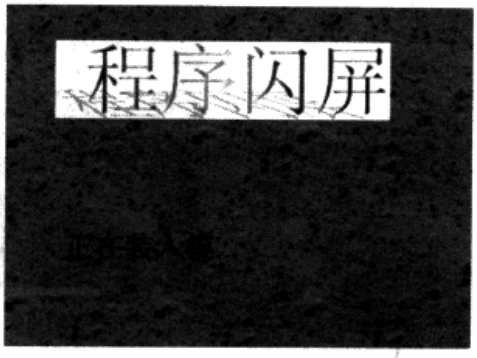


图 22-42 新闪屏的显示效果

22.5.7 分发应用程序

在富客户端程序开发完成后，就可以分发应用程序了。富客户端程序有两种分发方式：创建 ZIP 分发和创建 JNLP 应用程序。

如果右键单击模块套件节点，然后选择弹出菜单中的“生成 ZIP 分发”命令，则应用程序的可执行文件及其群集将打包在 ZIP 分发中。

如果右键单击模块套件节点，然后选择弹出菜单中的“生成 JNLP 应用程序”命令，则会将 JNLP 应用程序放在主机上。JNLP（Java Network Launch Protocol）为 Java 网络加载协议。JNLP 应用程序可以通过 Web Start 直接在客户端进行部署和运行。

如果要通过 NetBeans 的“更新中心向导”来分发，则要右键单击模块套件项目，然后选择弹出菜单中的“创建 NBM”命令，将为模块套件项目中的每个模块项目创建一个 NBM 文件和自动更新描述符。

22.6 小结

本章主要介绍 NetBeans 插件模块和富客户端应用程序的开发，包括创建操作并为操作编写代码、创建富客户端应用程序项目、为应用程序创建闪屏、分发模块和应用程序等内容。需要指出的是，本章对于 NetBeans 模块和富客户端应用程序的开发只是进行了入门的介绍，读者如果想要精通开发，需要进一步地了解如何使用各种不同的 NetBeans API。这里因为篇幅关系不再具体介绍，请有兴趣的读者参考 NetBeans 平台开发的相关书籍。

通过本章的学习，读者应该掌握以下基本内容：

- NetBeans 平台和 NetBeans 集成开发环境的关系
- 如何在模块套件项目中添加模块项目
- 如何为应用程序创建闪屏
- 如何分发应用程序和模块

Download at http://www.pin5i.com/

PART 5

第五篇　NetBeans 开发 C++程序

第 23 章　使用 NetBeans 开发 C/C++程序

Download at http://www.pin5i.com/

第23章

使用 NetBeans 开发 C/C++程序

NetBeans 不仅仅是一个功能强大的 Java 开发工具，也可以成为其他语言的开发平台。如果安装了具有 C/C++支持的 NetBeans 集成开发环境，还可以开发 C 和 C++应用程序，从而使 NetBeans 成为一款跨平台的 C/C++开发工具。带有 C/C++支持的 NetBeans 提供了 C/C++项目和模板、高级源代码编辑器、对编译器的支持以及 Makefile 和配置等功能，使开发 C/C++项目变得轻松快捷。

本章主要介绍了 NetBeans C/C++ Pack 软件包的安装和设置以及 C 和 C++项目的创建，最后着重讲解了利用 NetBeans C/++ Pack 软件包创建 JNI。

本章主要内容

- 安装和设置 NetBeans C/C++ Pack 软件包
- 创建和设置 C 和 C++项目
- 设置 NetBeans C/C++ Pack 软件包
- 利用 NetBeans C/C++软件包创建 JNI

本章亮点

- 安装并设置编译器和工具
- 创建 Makefile
- 利用 NetBeans C/C++ Pack 软件包创建 JNI

23.1 NetBeans C/C++ Pack 软件包简介

如今 C 和 C++语言依然非常流行，在一些领域 C 和 C++仍在大显身手。NetBeans 也不仅仅也一个 Java 集成开发环境，通过安装具有 C/C++支持的 NetBeans 集成开发环境，开发人员也可以开发出 C 和 C++应用程序。

NetBeans C/C++ Pack 软件包提供在 NetBeans 平台上开发 C 和 C++应用程序的功能。NetBeans C/C++ Pack 软件包提供了 C/C++项目的开发环境，具有代码完成、代码折叠以及语法突出显示功能的编辑器，对 Makefile 支持以及调试器功能。此外，NetBeans C/C++ Pack 软件包使用纯 Java 语言编写，可以运行在 Microsoft Windows XP、Linux、Solaris 10 以及 Mac OS X 操作系统等 NetBeans 支持的平台上。本章所讲解的内容全部以 Microsoft Windows XP 平台为例。

23.2 安装和设置 NetBeans C/C++ Pack 软件包

NetBeans C/C++ Pack 软件包需要在 Java SE Development Kit (JDK) 5 Update 14（版本 1.5.0_14）或更高版本（包括 JDK 6 Update 3）上运行，且与 NetBeans IDE 6.0 兼容，所以在安装 NetBeans C/C++ Pack 之前请确定已经安装了适合版本的 JDK 和 NetBeans IDE。

Download at http://www.pin5i.com/

23.2.1 下载安装具有 C/C++支持的 NetBeans 集成开发环境

可以在 NetBeans 网站 http://www.netbeans.org/下载具有 C/C++支持的 NetBeans 集成开发环境。选择具有 C/C++ Pack 的 NetBeans IDE 版本，单击“C/C++”或“All”下面的“Download”按钮即可。如果系统中已经安装没有 C/C++支持的 NetBeans 集成开发环境，可以使用下面的方法将 C/C++ Pack 安装到当前的 NetBeans 集成开发环境中。

❶ 如果用户网络使用代理，在 NetBeans 集成开发环境中，选择主菜单中的“工具”→“选项”命令，在打开的“选项”对话框中，选择“手动代理设置”单选按钮，分别在“HTTP 代理”文本框和“端口”文本框中输入 HTTP 代理和端口，单击“确定”按钮，如图 23-1 所示。

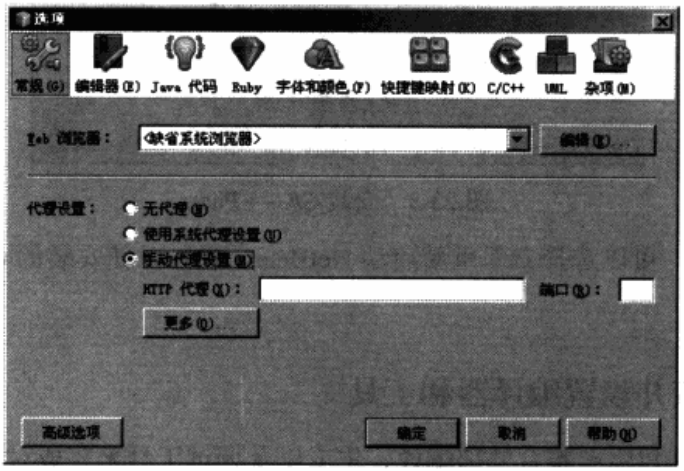


图 23-1 设置代理

❷ 从主菜单中选择“工具”→“插件”命令，在打开的“插件”对话框中选择“可用插件”标签，在插件列表中选择“C/C++”复选框，如图 23-2 所示。

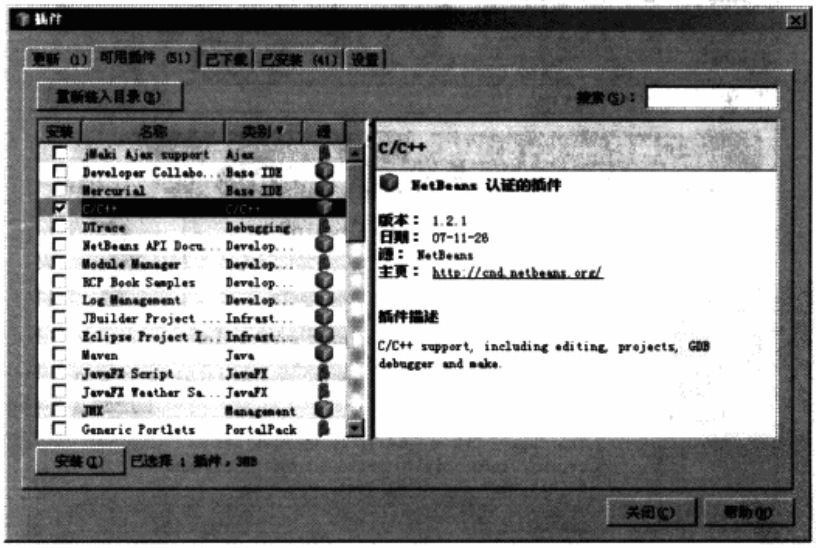


图 23-2 选择 C/C++

❸ 单击“安装”按钮，启动 NetBeans IDE 安装程序，单击“下一步”按钮。在打开的“NetBeans IDE 安装程序”对话框中选中“我接受所有许可证协议中的条款”单选按钮，单击“安装”按钮开始安装 C/C++ Pack，如图 23-3 所示。

Download at http://www.pin5i.com/

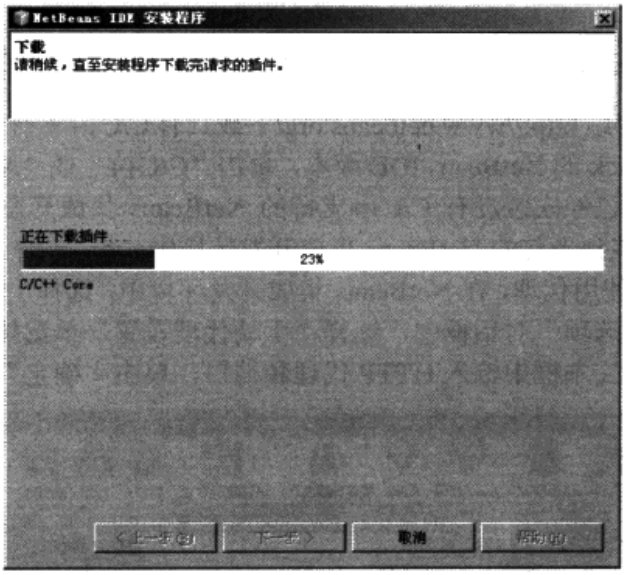


图 23-3 安装 C/C++ Pack

❹ 安装完成后，可以选择立即重新启动 NetBeans IDE 以使安装的 C/C++ Pack 生效，也可以稍后再重启。



23.2.2 安装并设置编译器和工具

C/C++ Pack 软件包需要整合外部编译、生成以及调试工具来完成对 C 和 C++项目的编译及调试等工作。这样开发人员可以根据不同的系统平台来选择适当的编译器等工具。

下面以 Cygwin 为例讲解编译器等工具的安装和设置。在 Cygwin 网站下载 Cygwin 的安装程序，下载页面如图 23-4 所示。

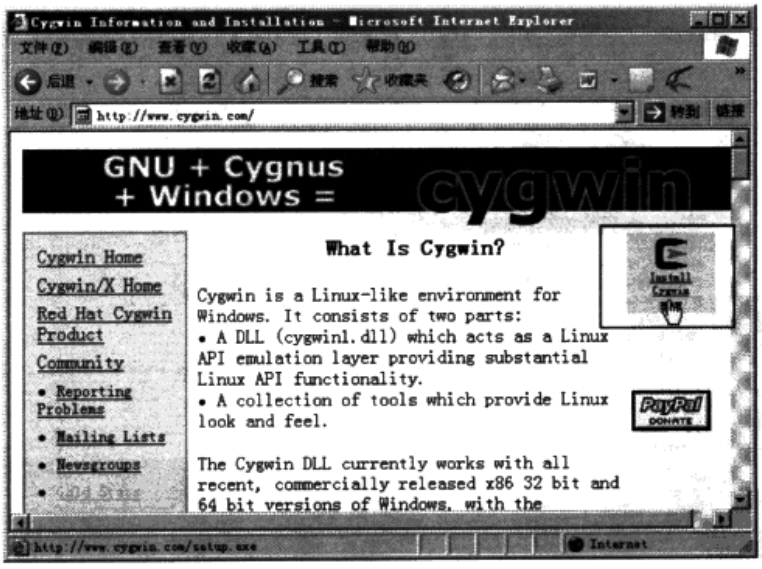


图 23-4 Cygwin 下载页面

下载完成后，双击安装文件 setup.exe，启动安装程序。

❶ 打开安装程序的欢迎窗口，单击“下一步”按钮。

❷ 打开“选择下载源”窗口，在这个窗口中可以选择三种下载方式，如图 23-5 所示。

Download at http://www.pin5i.com/

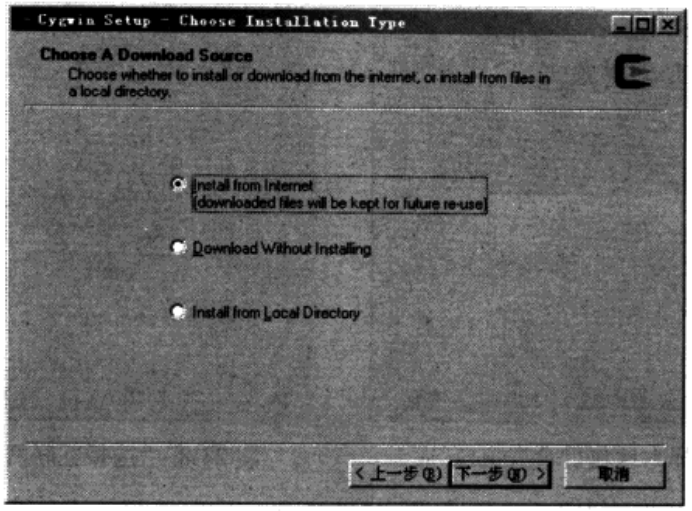


图 23-5 “选择下载源”窗口

- Install from Internet（从互联网上安装）：从互联网上下载安装文件，然后启动安装程序进行安装，下载后的安装文件可以供以后使用。
- Download Without Installing（仅下载安装文件而不安装）：将安装文件从互联网上下载下来但不启动安装程序。
- Install from Local Directory（从本地目录安装）：如果以前已经下载已安装程序，则可以使用此选项。

❸ 这里选择“Install from Internet”单选按钮，单击“下一步”按钮，打开“选择安装目录”窗口，如图 23-6 所示。在该对话框中可以设置 Cygwin 的安装目录，默认安装在 C:\cygwin 目录下。还可以选择是为用户自己安装还是为所有用户安装选项以及默认文本文件类型。

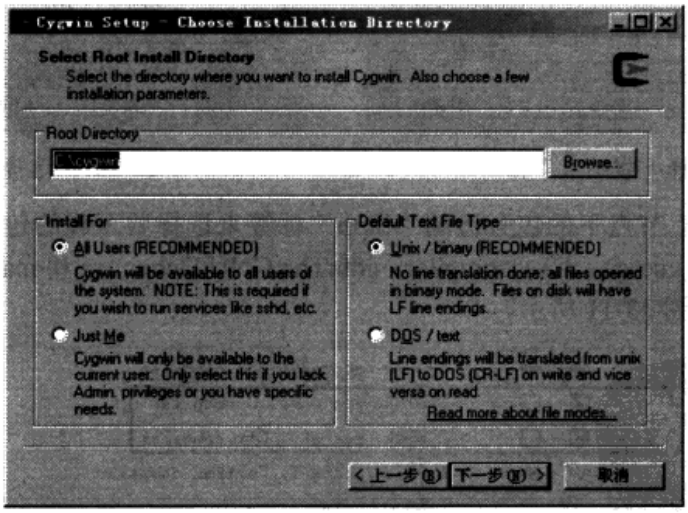


图 23-6 “选择安装目录”窗口

❹ 这里保持默认设置，单击“下一步”按钮，打开“选择本地包目录”窗口，如图 23-7 所示。在 Local Package Directory 文本框中输入，或者单击 Browse 按钮选择一个本地目录，用于存入下载以后安装程序文件。

❺ 单击“下一步”按钮，打开“选择互联网连接类型”窗口，如图 23-8 所示。这里可以选择 Direct Connection（直接连接）、Use IE5 Settings（使用 IE5 设置）以及 Use HTTP/FTP Proxy（使用 HTTP/FTP 代理）三个选项，选择“Direct Connection”单选按钮并单击“下一步”按钮。

Download at http://www.pin5i.com/

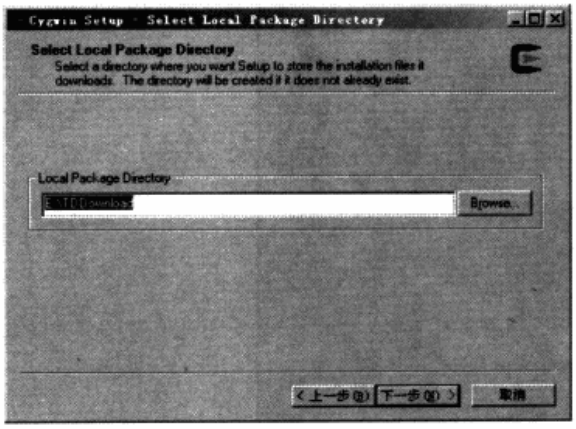


图 23-7 “选择本地包目录”窗口

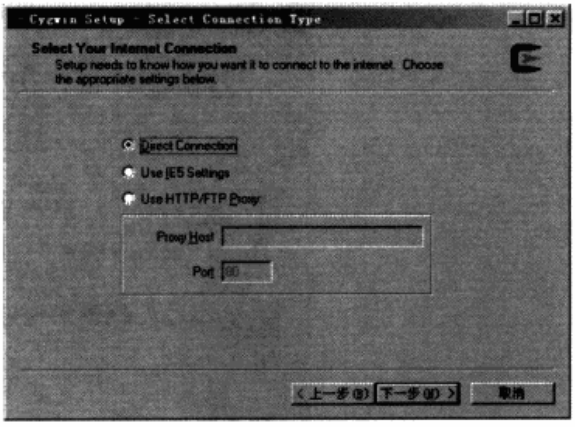


图 23-8 “选择互联网连接类型”窗口

❻ 打开“选择下载站点”窗口，如图 23-9 所示。这里选择一个距离自己最近的站点，单击“下一步”按钮。

❼ 打开“选择包”窗口，如图 23-10 所示。这里选择要下载的包，单击“Devel”左侧的“+”展开节点，为了一次看到更多的内容，最好将窗口最大化。

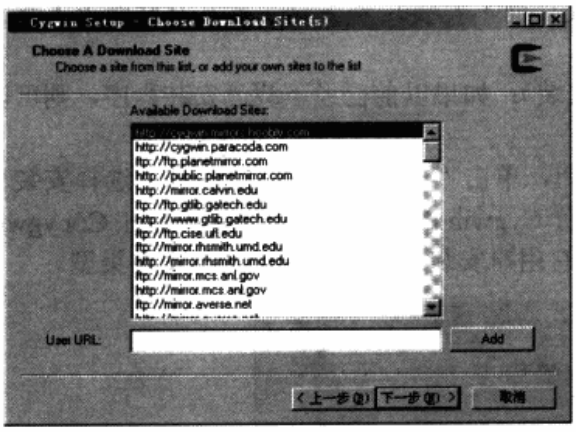


图 23-9 “选择下载站点”窗口

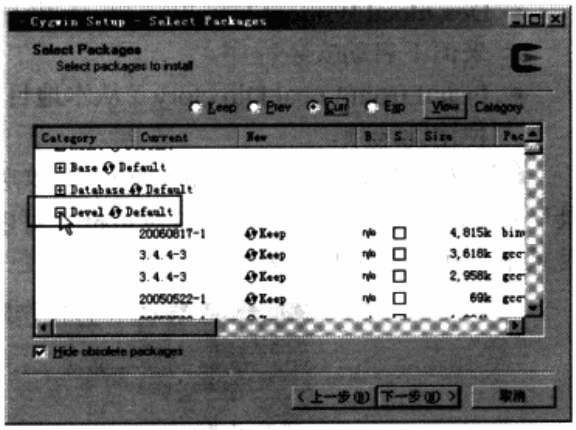


图 23-10 “选择包”窗口

通过单击 Devel 节点下的包旁边的“Skip”标签来选择要下载的包。在这里至少要选择 gcc-core:C compiler、gcc-g++:C++ compiler、gdb:The GNU Debugger 和 make:the GNU version of the 'make' utility，如图 23-11 所示。

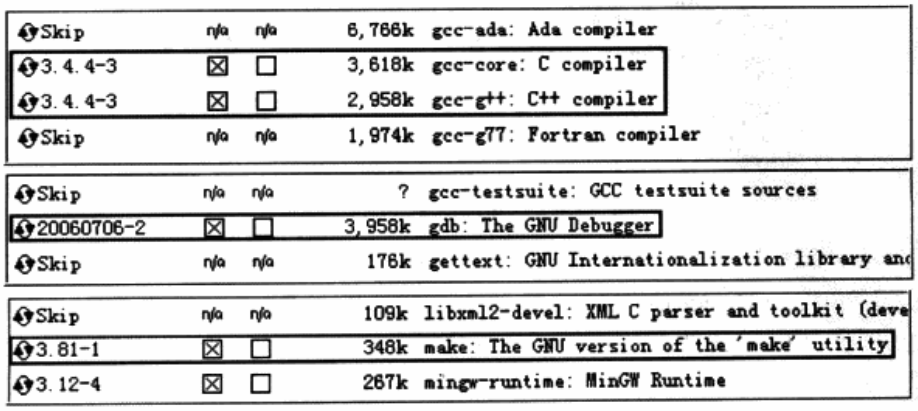


图 23-11 选择要下载的包

Download at http://www.pin5i.com/

❽ 安装程序开始复制文件，这可能需要一段时间。安装文件复制完成后，出现“创建图标”窗口，如图 23-12 所示。用户可以根据自己的需要选择 Create icon on Desktop（在桌面创建图标）或 Add icon to Start Menu（添加图标到开始菜单），最后单击“完成”按钮完成安装。

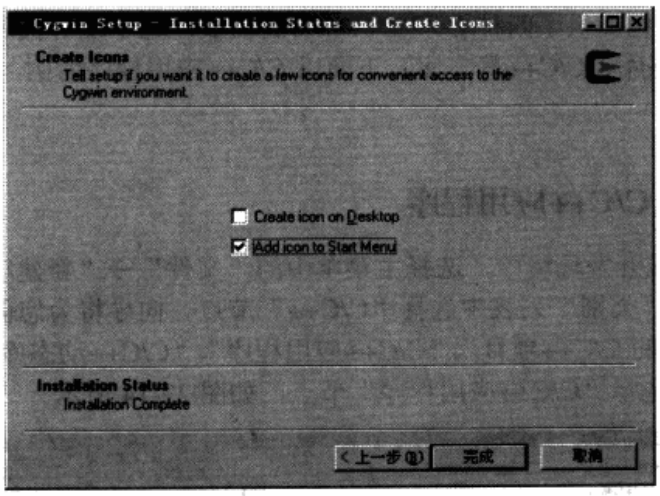


图 23-12 “创建图标”窗口

Cygwin 安装完成后还要将其目录添加到用户或系统的 Path 变量中。

❶ 在桌面上右键单击“我的电脑”，选择弹出菜单中的“属性”。

❷ 在打开的“系统属性”对话框中选择“高级”标签，然后单击“环境变量”按钮。

❸ 在“环境变量”对话框的“用户变量”面板或“系统变量”面板中，选择“Path”变量，然后单击“编辑”按钮。

❹ 将 C:\cygwin\usr\bin 和 C:\cygwin\bin 目录的路径添加到 Path 变量中，然后单击“确定”按钮，如图 23-13 所示。目录名称之间必须用分号进行分隔。

❺ 在“环境变量”和“系统属性”对话框中分别单击“确定”按钮来完成环境变量的设置。

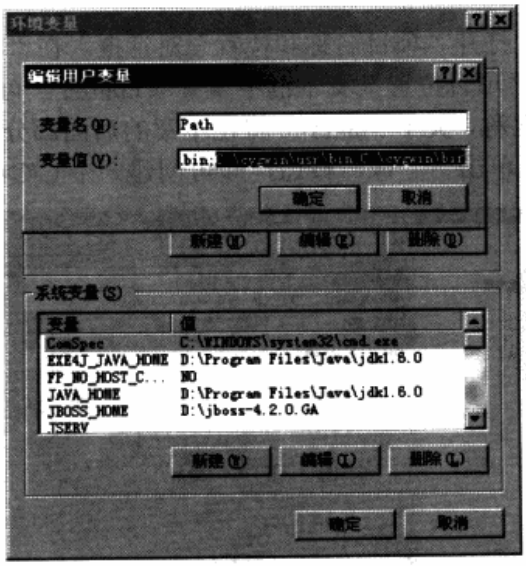


图 23-13 设置环境变量

至此，编译器和工具的安装和设置工作到此结束，下面可以进入 C 和 C++项目的开发。

Download at http://www.pin5i.com/

23.3 创建和设置 C 和 C++项目

C/C++ Pack 软件包提供了四种 C 和 C++项目的模板：基于现有代码的 C/++项目、C/C++应用程序、C/C++动态库、C/C++静态库。下面以 C/C++应用程序为例，讲解 C 和 C++项目的创建和设置。

23.3.1 创建 C/C++应用程序

在 NetBeans 集成开发环境中，选择主菜单中的“文件”→“新建项目”命令，打开“选择项目”对话框。在“类别”列表中选择“C/C++”节点。向导将为您创建新项目提供的四种选项：“基于现有代码的 C/C++项目”、“C/C++应用程序”、“C/C++动态库”和“C/C++静态库”。这里“项目”列表中选择“C/C++应用程序”节点，如图 23-14 所示。

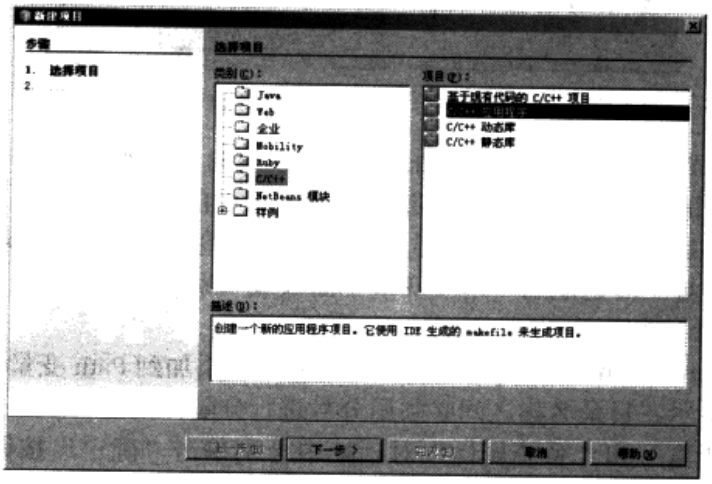


图 23-14 “选择项目”对话框

单击“下一步”按钮，打开“项目名称和位置”对话框。在“项目名称”文本框中输入项目名称“HelloWorld”，在“项目位置”文本框中设置项目位置。可以在“项目 Makefile 名称”文本框中输入 Makefile 的名称，关于 Makefile 将在本章的后面部分进行介绍，其他选项保持默认设置，如图 23-15 所示。单击“完成”按钮，即可创建一个 C/C++应用程序项目。

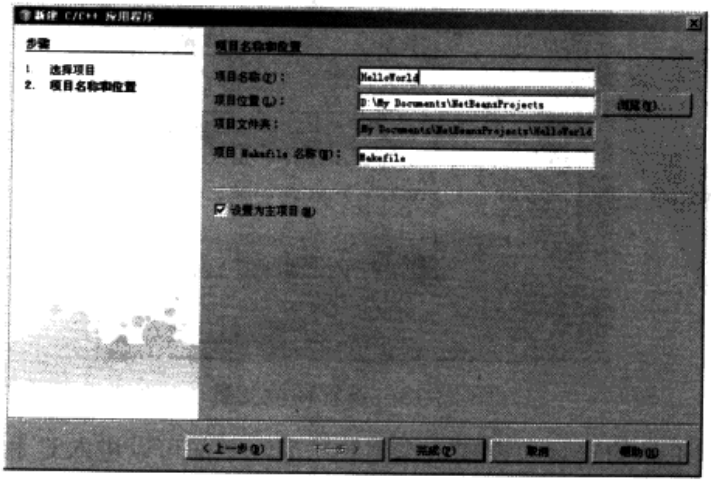


图 23-15 “项目名称和位置”对话框

Download at http://www.pin5i.com/

23.3.2 逻辑视图和物理视图

一个项目具有逻辑视图和物理视图两种视图方式。“项目窗口”中的视图方式为逻辑视图方式，逻辑视图中文件夹不是目录，只是一种组织文件的方式，不代表文件在磁盘上的物理存储位置。HelloWorld 项目的逻辑视图如图 23-16 所示。

说明：“重要文件”文件夹中的文件不是项目的一部分，在生成项目时不会编译这些文件。如 Makefile 就存放在这个文件夹中。

“文件”窗口中的视图方式为物理视图。该视图中的文件结构和物理磁盘上存储的文件和文件夹的组织结构是一样的，如图 23-17 所示。



图 23-16 逻辑视图

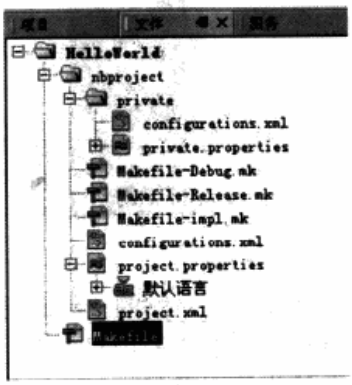


图 23-17 物理视图

23.3.3 向项目中添加文件

在项目中新建文件。在“项目”窗口中，右键单击“源文件”节点，选择弹出菜单中的“新建”→“文件/文件夹”命令。在打开的“选择文件类型”对话框中选择“主 C 文件”，如图 23-18 所示。单击“下一步”按钮。

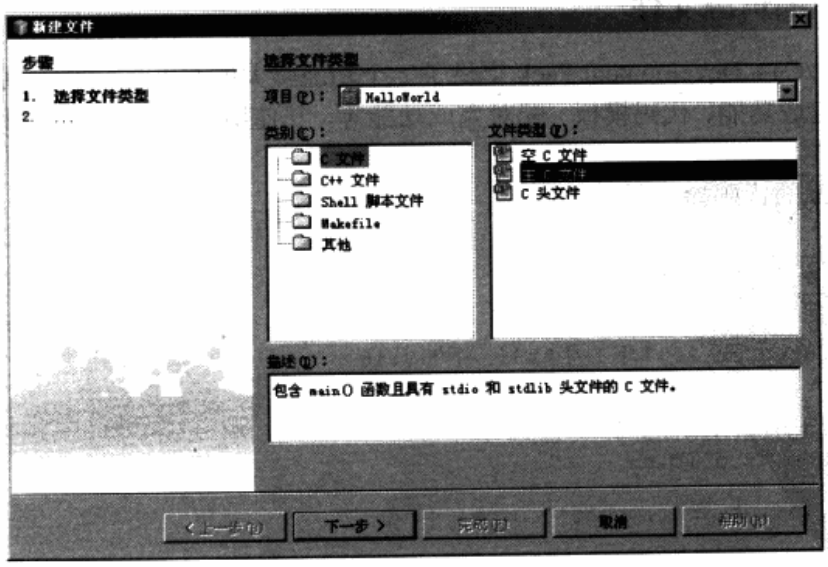


图 23-18 “选择文件类型”对话框

Download at http://www.pin5i.com/

在“名称和位置”对话框中输入文件名称 hello，单击“完成”按钮。这样就新建了一个源文件。

还可以将已经存在的文件添加到项目中。在“项目”窗口中，右键单击 HelloWorld 项目节点，选择弹出菜单中的“添加现有项”命令，在打开的“选择项”对话框中选择要添加的文件，如图 23-19 所示。

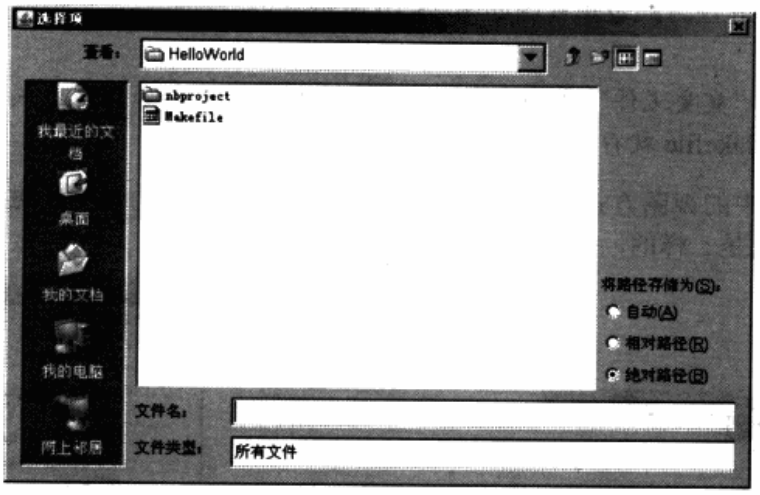


图 23-19 “选择项”对话框

这里对对话框右侧的“将路径存储为”单选按钮组进行一下说明。

- 自动：如果位于项目内部，则将路径存储为相对于项目根目录的相对路径；否则，将其存储为绝对路径。
- 相对路径：始终将路径存储为相对于项目根目录的相对路径。
- 绝对路径：始终将路径存储为绝对路径。

如果在弹出的菜单中选择“从文件夹中添加现有项”命令，可以将指定文件夹中的源文件添加到项目中；选择“新建逻辑文件夹”命令，可以建立新的逻辑文件夹；选择“将项添加到重要文件中”命令，则可以将外部项添加到重要文件逻辑文件夹中。

23.3.4 编辑源文件

NetBeans C/C++ Development Pack 提供了一个功能强大的 C 和 C++语言源代码编辑器，其功能包含代码完成功能、代码模板、配对完成功能等，其使用方法和 Java 的源代码编辑基本一致，这里不再赘述。

修改 hello.c 的代码如下所示：

```
/*
 * File:   hello.c
 * Author: admin
 *
 * Created on 2007年7月21日, 下午9:46
 */

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv) {

    int i;

    for (i = 1; i < argc; i++) {
```

Download at http://www.pin5i.com/


```
            printf("Hello: %s\n",argv[i]);//打印输出参数
        }
        return (EXIT_SUCCESS);
    }
```

1. 使用类视图

C/C++ Pack 提供了用于查看 C++项目中的所有类，以及每个类的成员和字段的类视图，如图 23-20 所示。要打开类视图，选择主菜单中的“窗口”→“类视图”命令。在项目节点下列出了项目中所有的类，展开某个类，可以查看其字段、方法列表等信息，双击元素节点可以转到该元素声明。

2. 使用导航窗口

导航窗口提供当前选择文件的一个紧凑的结构视图，并且可以在文件的不同部分进行导航。在编辑器窗口、项目窗口或者文件窗口中选择某个文件，该文件的结构就会显示在导航窗口中，如图 23-21 所示。

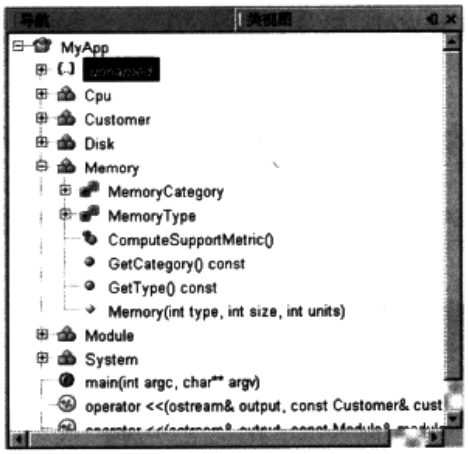


图 23-20 类视图

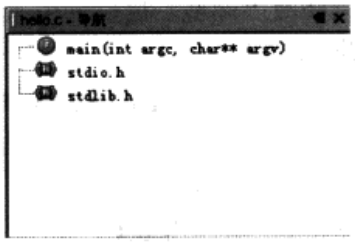


图 23-21 导航窗口

在导航窗口中，双击文件中的某个元素，则编辑器窗口中的光标就会移动到当前元素的位置。

23.3.5 设置项目属性

创建 C 或 C++项目后，该项目将具有调试（Debug）和发行（Release）两种配置。通过对不同配置的项目属性的设置从而生成不同版本的项目。可以在“项目属性”对话框中修改编译器设置、链接器设置以及其他配置属性。

在“项目”窗口中右键单击 HelloWorld 项目节点，然后选择弹出菜单中的“属性”命令，打开“项目属性”对话框，如图 23-22 所示。在“配置”下拉列表框中选择需要修改的配置。下面对一些重要常用的属性进行说明，其中一些功能类似的属性不再重复介绍，请读者参考相关属性的说明。

1. 设置常规属性

选择“类别”列表下的常规节点，在右侧的属性列表中列出了项目的常规信息。可以在“编译器集合”下拉列表框中选择当前项目使用的编译器集合，这里只有一个选项“来自 Cygwin 的 GNU 编译器集合”。在“配置类型”下拉列表框中选择项目的配置类型，即当前是应用程序、动态库、静态库还是 Makefile。

Download at http://www.pin5i.com/

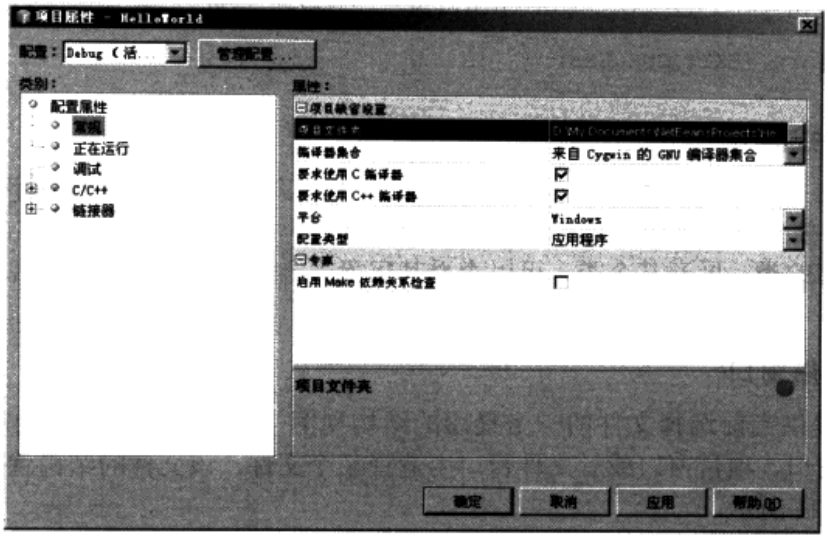


图 23-22　设置常规属性

2. 设置正在运行属性

选择“正在运行”节点，如图 23-23 所示。修改“参数”属性的值为“tom bob jim ketty”。这个属性用于设置 main()函数的参数。“运行目录”属性用于设置执行二进制文件时使用的目录。可以修改“环境”属性的值来为程序运行添加环境变量。选中“首先生成”复选框可以在执行二进制文件之前生成项目。

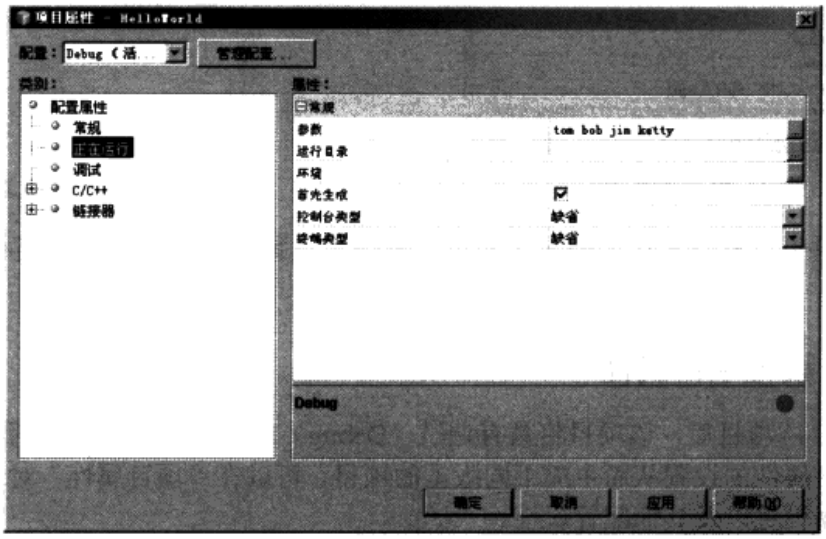


图 23-23　设置正在运行属性

3. 设置 C 编译器常规属性

依次展开节点“C/C++”→“C 编译器”→“常规”，如图 23-24 所示。通过添加“包含目录”可以将项目编译时所需要的头文件所在的目录包含到项目中，这样编译器在编译时可以找到相应的头文件。可以设置“预处理程序定义”的值来为项目定义预处理程序。

4. 设置 C 编译器命令行属性

选择“命令行”节点，如图 23-25 所示。通过设置“其他选项”属性的值来为编译器添加编译时的命令行参数，这些参数对于不同的编译器是不一样的。

Download at http://www.pin5i.com/

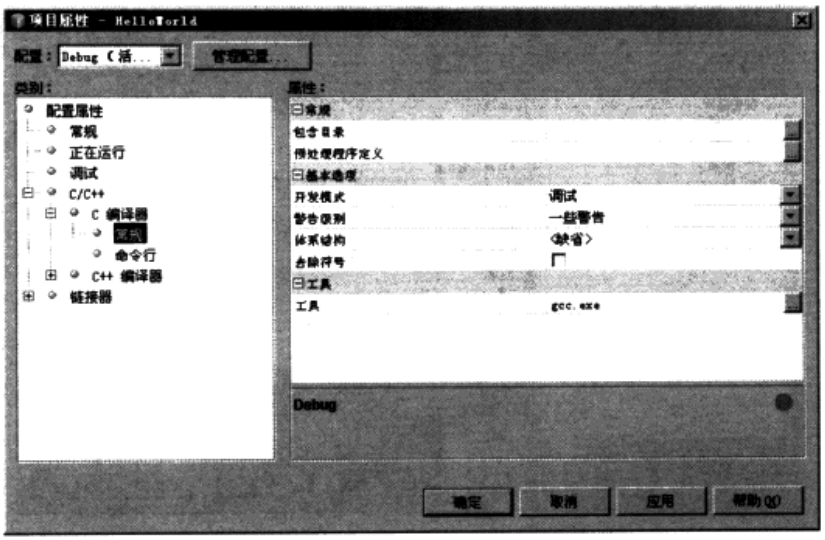


图 23-24　设置 C 编译器常规属性

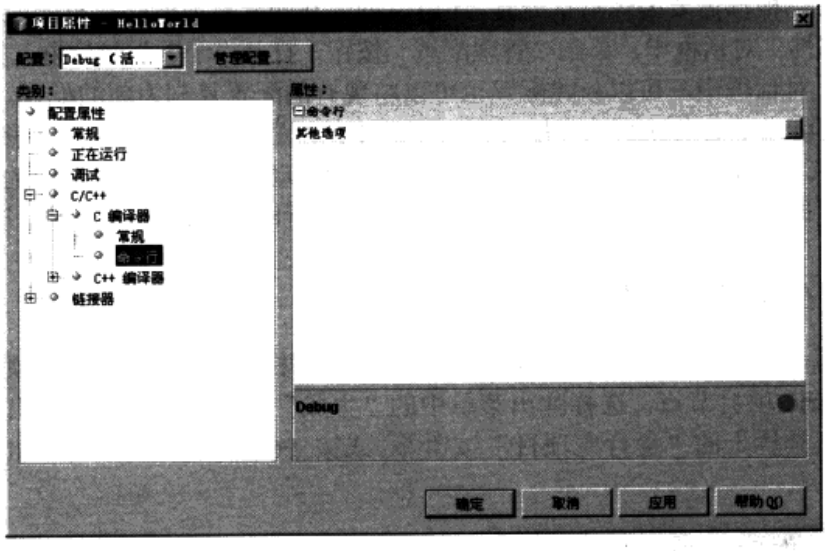


图 23-25　设置编译器命令行属性

C++编译器的常规和命令行的属性和 C 编译器的属性是一样的，这里就不再赘述了。

5. 设置链接器常规属性

展开节点“链接器”→“常规”，如图 23-26 所示。设置“输出”属性的值可以指定项目的输出路径。可以添加“其他库目录”属性来为链接器设定其他库的目录。

另外，还可以对单个文件的属性进行设置。在“项目”窗口中，右键单击选择设置属性的文件节点，选择弹出菜单中的“属性”命令。然后在打开的“项目属性”对话框中设置文件的属性，文件属性的值会自动继承项目相应属性的设置值。对于单个文件，为其设置的编译器配置及属性将覆盖项目编译器的设置和其他属性。也可以在特定的配置中设置排除生成某些文件。

23.3.6　管理配置

配置是用来表示项目存储当前设置的方式。在项目具有的两个配置中，使用“Debug”配置可以生成包含调试信息的应用程序版本，使用“Release”配置可以生成一个优化的用于发行的版本。

Download at http://www.pin5i.com/

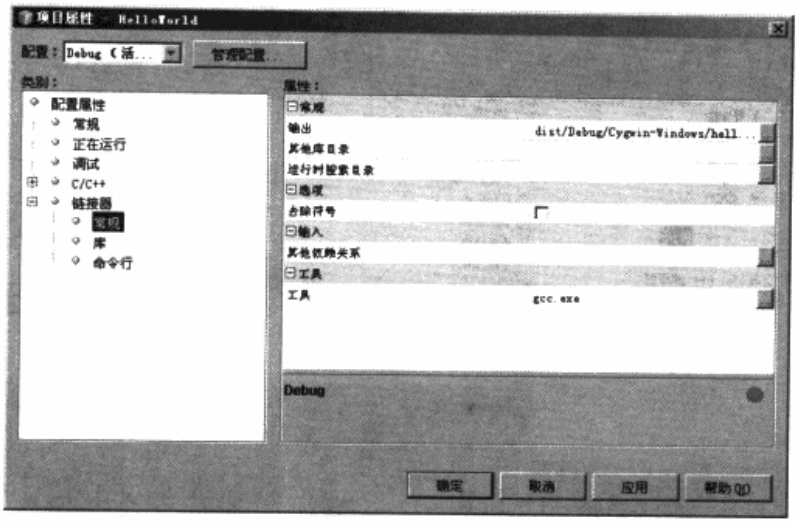


图 23-26 设置链接器常规属性

开发人员可以根据需要编辑默认配置，还可以创建新的配置或删除已有的配置。要管理配置，可以在“项目属性”对话框中，单击“管理配置”按钮，打开“配置”对话框，如图 23-27 所示。

在“配置”对话框中，可以新建配置也可以将现有的配置复制为新的配置。单击“重命名”或“删除”按钮可以重命名或删除在配置列表中选中的配置，也可以单击“上移”或“下移”按钮来改变配置在“配置”列表中的顺序。单击“设置为活动”按钮可以将选中的配置设置为活动配置，活动配置即为当前使用的配置，在项目生成和运行时将使用活动配置中设定的属性值。

23.3.7 生成并运行项目

在代码编写以及项目属性设置完成之后，可以生成并运行项目了。在“项目”窗口中，右键单击 HelloWorld 项目节点，选择弹出菜单中的“生成”命令，可以在“输出”窗口中看到生成信息。单击工具栏上的“运行主项目”按钮 或按 F6 键运行 HelloWorld，运行结果如图 23-28 所示。

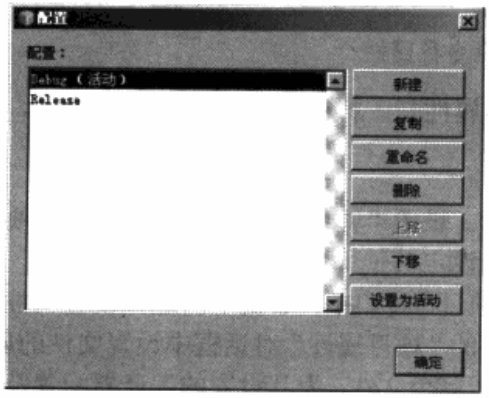


图 23-27 “配置”对话框



图 23-28 项目运行结果

23.4 设置 C/C++ Pack

在 NetBeans 集成开发环境中，可以对 C/C++ Pack 进行设置。要设置 C/C++ Pack，选择主菜单中的“工具”→“选项”命令，打开“选项”对话框，然后选择“C/C++”标签。

Download at http://www.pin5i.com/

23.4.1 设置 C/C++生成工具

在“选项”对话框中选择“C/C++”标签下的“生成工具”标签，如图 23-29 所示。

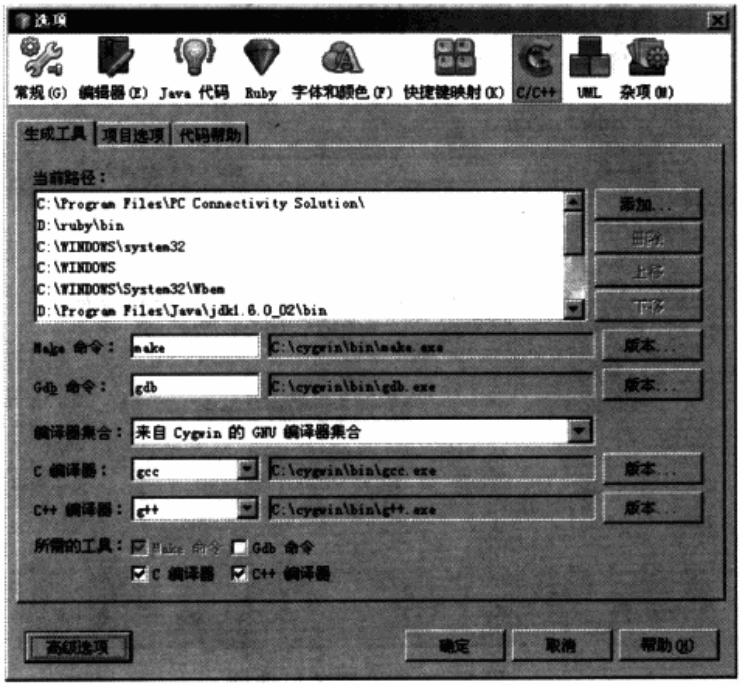


图 23-29 设置生成工具

在此对话框中，可以对与生成工具相关的属性进行设置，具体的属性设置说明如表 23-1 所示。

表 23-1 生成工具设置说明

属性名称	说 明
当前路径	在“当前路径”列表中列出了集成开发环境使用的路径列表，NetBeans C/C++ Development Pack 用此路径列表中的路径查找所需的生成工具。可以单击右侧的“添加”或“删除”按钮来添加新的路径或删除选中的路径，还可以单击“上移”或“下移”按钮来改变路径在列表中的前后顺序
make 命令	生成项目时使用的 make 命令。默认命令为当前路径中找到的第一个 make 命令。如果你要更改该命令，则系统会在当前路径中搜索新的 make 命令。单击“版本”按钮可以显示当前 make 命令的版本信息
gdb 命令	调试项目时使用的 gdb 命令。默认命令为当前路径中找到的第一个 gdb 命令。如果你要更改该命令，则系统会在当前路径中搜索新的 gdb 命令。单击“版本”按钮可以显示当前 gdb 命令的版本信息
编译器集合	生成项目时使用的编译器集合。系统会在当前路径中搜索包含编译器的目录，并将其添加到该下拉列表框中
C 编译器	生成项目时使用的 C 编译器。默认编译器为选定编译器集合中找到的第一个编译器。单击“版本”按钮可以显示当前 C 编译器的版本信息
C++编译器	生成项目时使用的 C++编译器。默认编译器为选定编译器集合中找到的第一个编译器。单击“版本”按钮可以显示当前 C++编译器的版本信息
所需的工具	生成项目所需的工具。如果要调试项目，则要选中“Gdb”复选框，要编译 C 源文件或 C++源文件要分别选中“C 编译器”或“C++编译器”复选框。

Download at http://www.pin5i.com/

23.4.2 设置 C/C++项目选项

在“选项”对话框中选择“C/C++”标签中的“项目选项”标签，如图 23-30 所示。

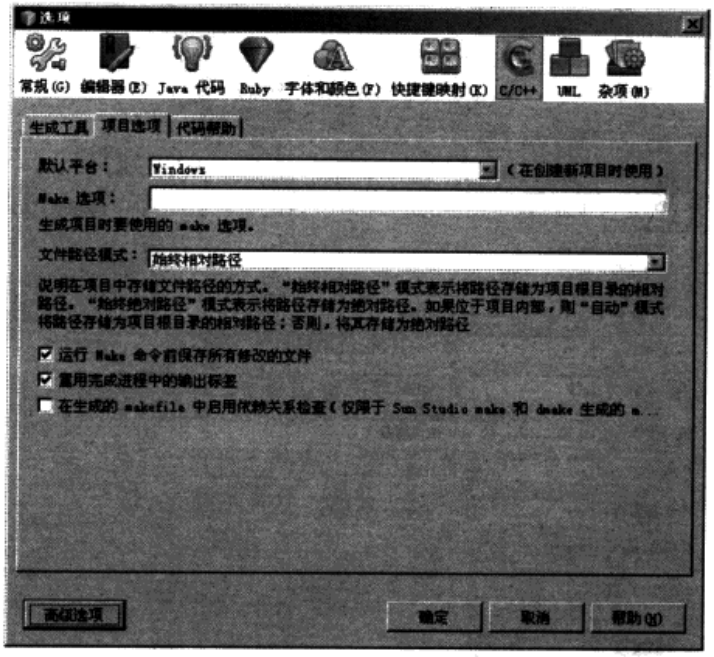


图 23-30 设置项目选项

在此对话框中，可以对 C 和 C++项目的默认选项进行设置，具体的属性设置说明如表 23-2 所示。

表 23-2 项目选项设置说明

属性名称	说 明
默认平台	创建新项目时使用的默认平台。系统会自动检测，一般不需要修改
Make 选项	make 命令的全局选项
文件路径模式	在项目中存储文件路径的模式。“始终相对路径”模式表示将路径存储为相对于项目根目录的相对路径。“始终绝对路径”模式表示将路径存储为绝对路径。如果位于项目内部，则“自动”模式将路径存储为项目根目录的相对路径；否则，将其存储为绝对路径
运行 Make 命令前保存所有修改的文件	如果选中此复选框，则会在运行 make 前保存集成开发环境中所有未保存的文件。建议选中此复选框，如果不首先将集成开发环境中的文件修改保存到磁盘上，make 命令将无法识别这些修改
重用完成进程中的输出标签	如果选中此复选框，则会将 make 输出写入到单个“输出”窗口标签中，而删除上一个进程的输出。如果未选中此属性，则会为每个 make 进程打开一个新标签
在生成的 Makefile 中启用依赖关系检查	在 Makefile 中添加 make 状态语句。仅限于使用 Sun Studio make 和 dmake 生成的 Makefile

23.4.3 设置 C/C++代码帮助

在“选项”对话框中选择“C/C++”标签中的“代码帮助”标签，如图 23-31 所示。

在此对话框中，可以对用于代码帮助功能的内置解析器进行设置。首先，在“编译器集合”下拉列表框中选择要设置的编译器集合，这里只有一个“来自 Cygwin 的 GNU 编译器集合”，然后选择相应编译器标签，如“C 编译器”或“C++编译器”。

Download at http://www.pin5i.com/

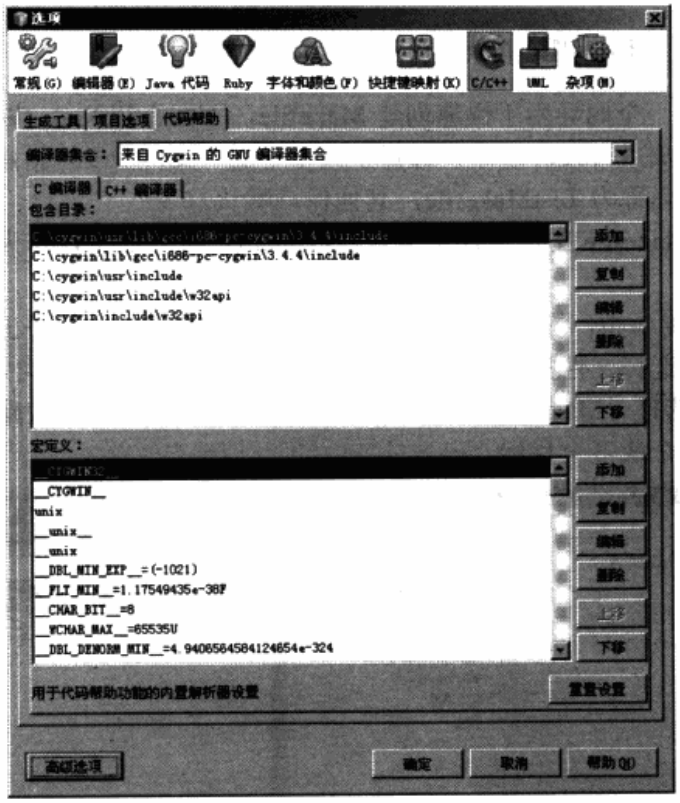


图 23-31 设置代码帮助

在“包含目录”列表中列出了内置解析器使用的包含文件所在目录，可以对其进行添加、复制、编辑以及删除操作。同样，可以对提供的宏定义进行相应的操作。单击“重置设置”按钮可以将当前标签中的设置重置为原始的默认设置。

23.5 创建 Makefile

集成开发环境通过 Makefile 控制项目的生成和运行的操作，本节主要讲解使用 C/C++ Pack 提供的 Makefile 向导来创建 Makefile。

23.5.1 Makefile 简介

Makefile 是 make 实用程序使用的 shell 命令文件。通常，make 实用程序使用 Makefile 来确定哪些程序部分需要编译以及用于其他方面。

C 和 C++项目系统是直接构建在 make 实用程序基础之上的。所有项目命令（如“生成”或“运行”）均调用项目 Makefile 中的目标。

使用集成开发环境开发人员无须了解 make 实用程序。对于 C/C++应用程序、动态库或静态库项目，可以在项目的“项目属性”对话框中设置所有基本编译和运行时的选项，集成开发环境会自动将更改的属性存储在当前配置的 Makefile 中。开发人员可以使用集成开发环境提供的 Makefile 向导来创建 Makefile，也可以自己手动编写 Makefile。

对于基于现有 Makefile 的 C/C++项目，在集成开发环境中设置项目属性时不会更新 Makefile。如果要更改在集成开发环境外部创建的 Makefile，必须手动修改 Makefile。

Download at http://www.pin5i.com/

23.5.2 使用 Makefile 向导创建 Makefile

NetBeans 提供了一个向导用于快速创建 Makefile，下面介绍 Makefile 文件的创建。

❶ 在 NetBeans 集成开发环境中，选择主菜单中的“文件”→“新建项目”命令，新建一个 C/C++应用程序，名称为 TestMakefile，其他保持默认。

❷ 在项目 TestMakefile 中，新建一个名称为 main.c 的主 C 文件作为项目源文件，不需要编辑其代码。

❸ 在“项目”窗口中，右键单击 TestMakefile 项目节点，选择弹出菜单中的“新建”→“其他”命令，打开“选择文件类型”对话框，如图 23-32 所示。展开“类别”列表中的 Makefile 节点，“文件类型”选择“Makefile”。这样可以创建新的 Makefile 以构建应用程序、静态库或动态库，可以定义一个或多个目标。

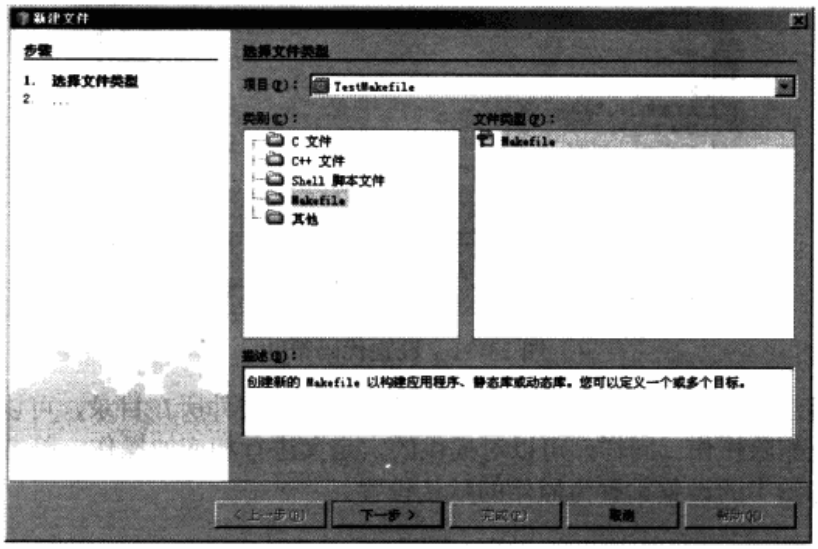


图 23-32 “选择文件类型”对话框

❹ 单击“下一步”按钮，打开“名称和位置”对话框，如图 23-33 所示。指定 Makefile 的文件名和所在的文件夹，这里在“文件名”文本框中输入文件名 MyMakefile。

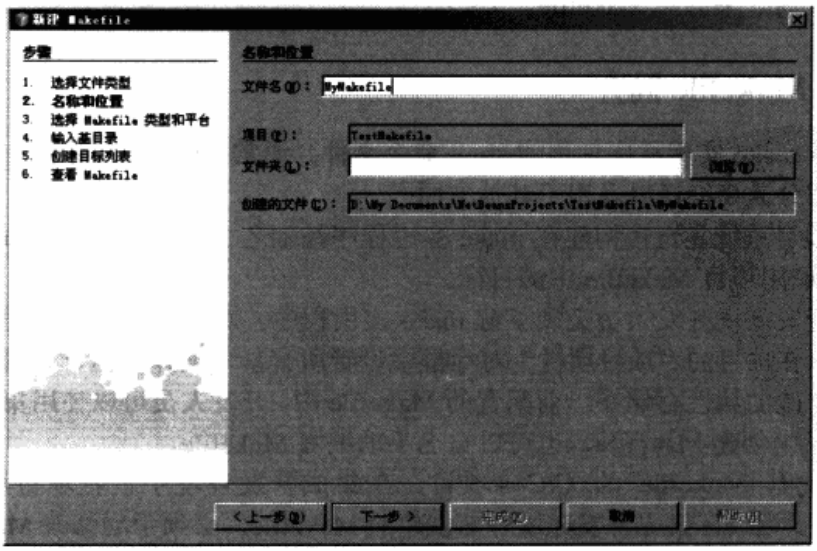


图 23-33 “名称和位置”对话框

Download at http://www.pin5i.com/

❺ 单击“下一步”按钮，打开“选择 Makefile 类型和平台”对话框，如图 23-34 所示。选择要使用的编译器集合和要运行 Makefile 所生成的二进制文件的平台。每个平台上的链接器标志和系统库都是不同的。

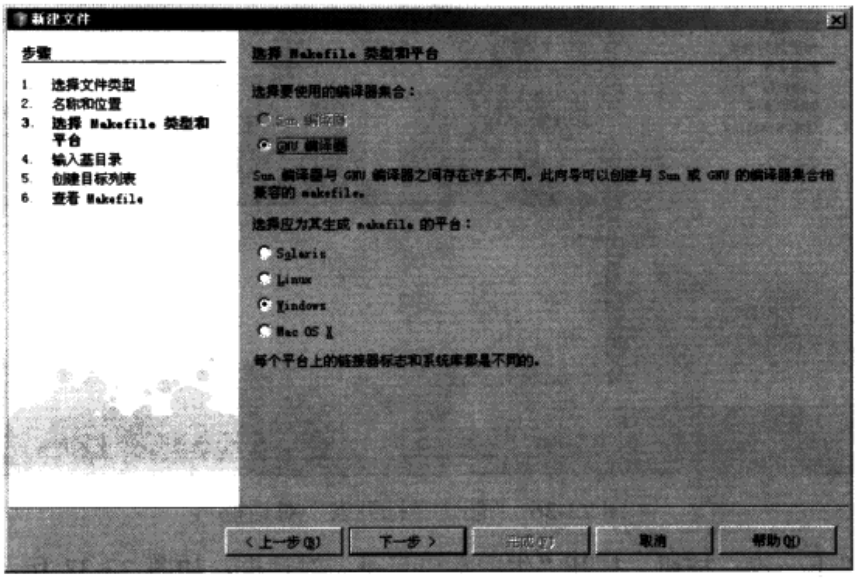


图 23-34 “选择 Makefile 类型和平台”对话框

❻ 这里保持默认设置，单击“下一步”按钮，打开“输入基目录”对话框，如图 23-35 所示。可以在“基目录”文本框中指定集成开发环境执行 make 命令所在的目录。通常情况下，基目录是 Makefile 和源文件树所在位置的路径。

❼ 这里保持默认设置，单击“下一步”按钮，打开“创建目标列表”对话框，如图 23-36 所示。使用该对话框可以在 Makefile 中指定要包含的目标，目标就是指需要将 Makefile 生成的内容。

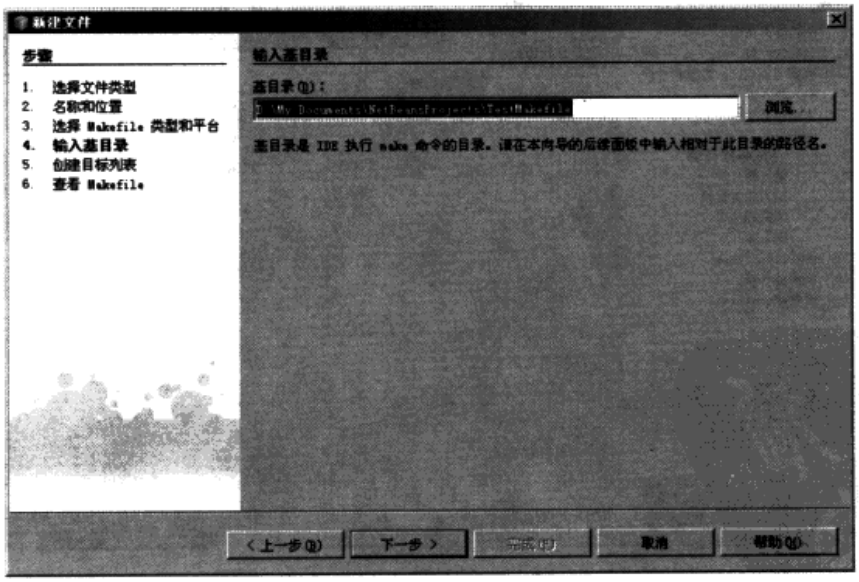


图 23-35 “输入基目录”对话框

这里创建两个目标。在“目标名称”文本框中输入 build 并选择“可执行文件”单选按钮，按回车按钮或单击“添加”按钮即可将 build 目标添加到 Makefile 目标列表中。同样，在“目标名称”文本框中输入 show 并选择“定制目标”单选按钮，然后单击“添加”按钮来添加 show 目标。

Download at http://www.pin5i.com/

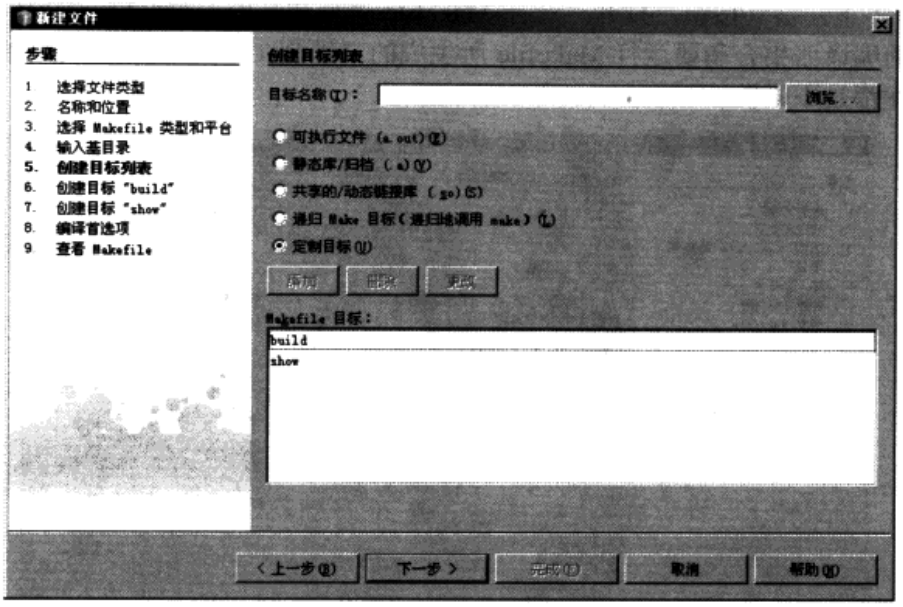


图 23-36 “创建目标列表”对话框

❽ 单击“下一步”按钮，打开“生成输出目录”对话框，如图 23-37 所示。在此对话框中可以指定用来存储目标所创建文件的目录。

❾ 这里保持默认设置，单击“下一步”按钮，打开“输入源文件”对话框，如图 23-38 所示。在此对话框中指定用于生成目标的源文件。单击“浏览”按钮，在打开的“选择文件”对话框中选择前面新建的 main.c 源文件，main.c 会被添加到“源文件”列表中。也可以在文本框中输入源文件的路径并单击“添加”按钮将文件添加到“源文件”列表中。

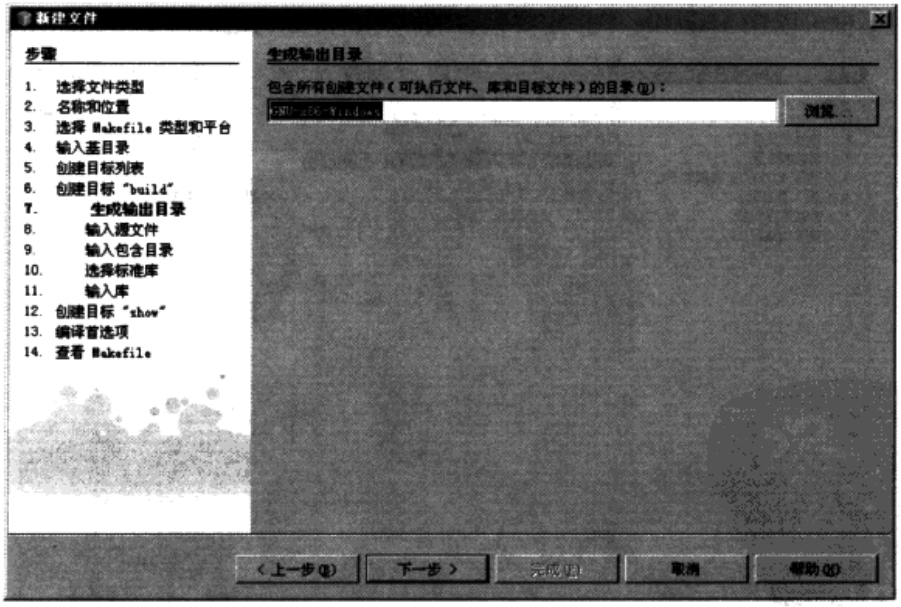


图 23-37 “生成输出目录”对话框

❿ 单击“下一步”按钮，打开“输入包含目录”对话框，如图 23-39 所示。在该对话框中可以指定希望编译器在哪些目录中搜索包含文件。可以在“要搜索包含文件的目录”文本框中输入目录并单击“添加”按钮或单击“浏览”按钮通过“选择文件”对话框来选择将目录添加到“包含目录”列表中。

Download at http://www.pin5i.com/

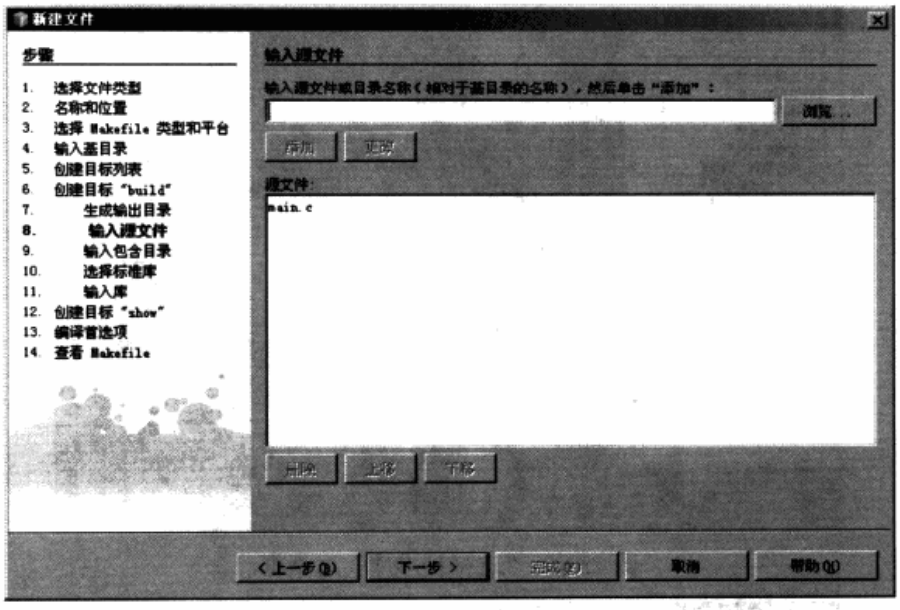


图 23-38 “输入源文件”对话框

⓫ 这里不需要添加任何目录，直接单击“下一步”按钮，打开“选择标准库”对话框，如图 23-40 所示。可以在该对话框中指定一些希望编译器将其与目标链接在一起的标准库。

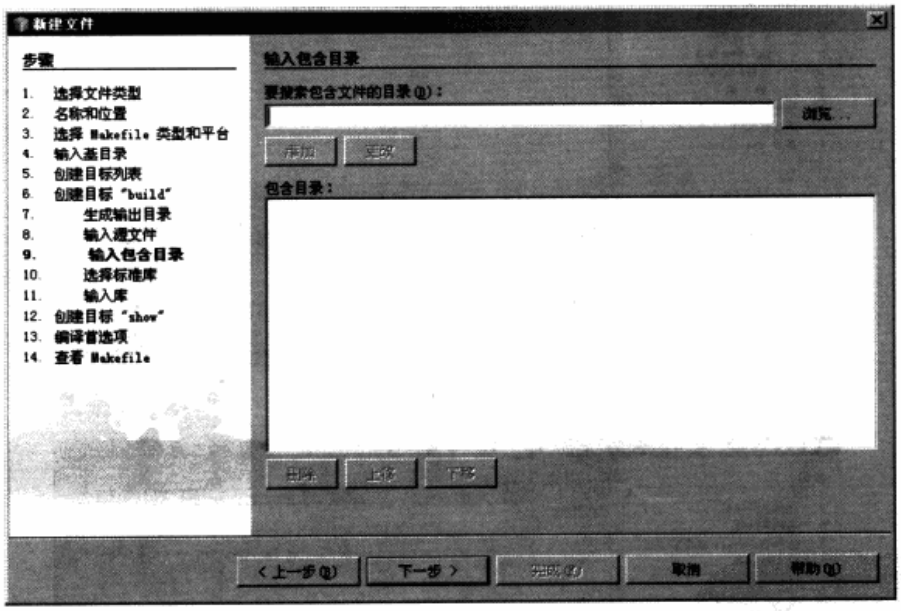


图 23-39 “输入包含目录”对话框

⓬ 这里不需要选择，单击“下一步”按钮，打开“输入库”对话框，如图 23-41 所示。可以在该对话框中指定一些希望编译器将其与目标链接在一起的库。“库”列表包含了一个 SYSLIBS 宏，该宏含有你在“选择标准库”页中选择的标准库值。这里不需要添加库，单击“下一步”按钮。

⓭ 以上步骤用于创建目标 build，这一步用于定义目标 show。出现“定义定制的 Make 目标”对话框，如图 23-42 所示。可以在“依赖于”文本框中输入当前目标依存的目标，这里保持为空。在“操作”文本域输入@echo “Show me the way!”，这里只是简单地输出一些信息。

Download at http://www.pin5i.com/

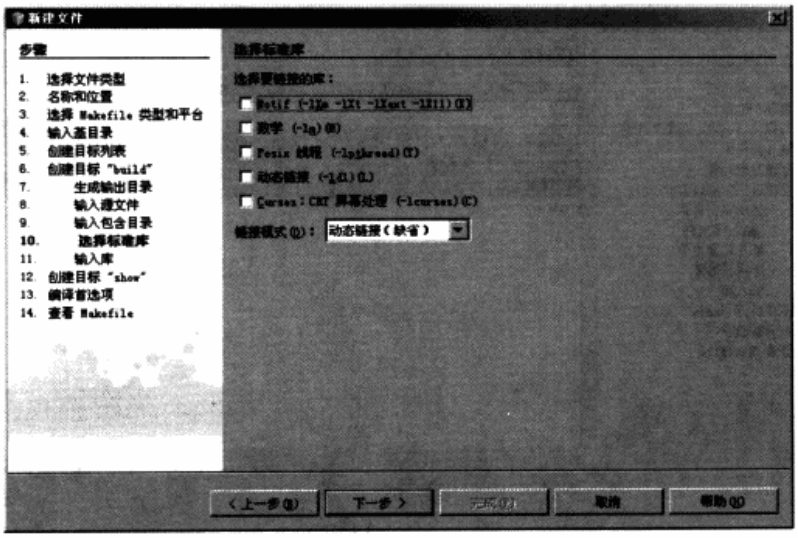


图 23-40 “选择标准库”对话框

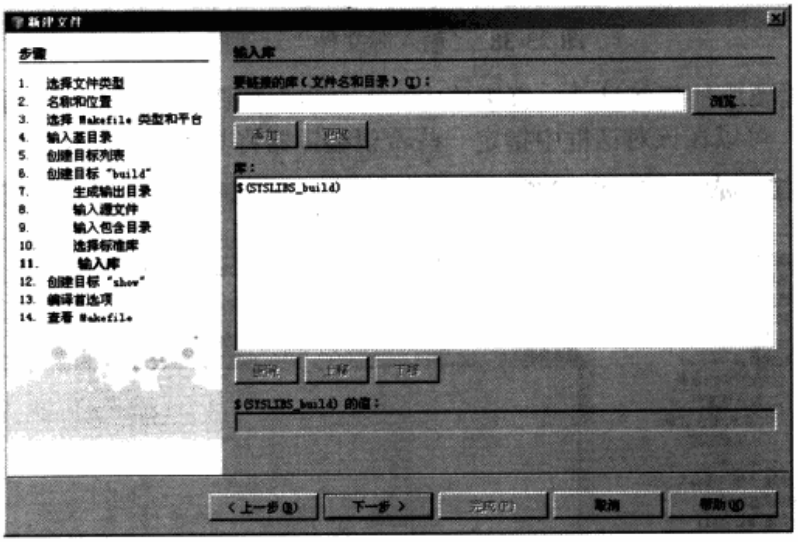


图 23-41 “输入库”对话框

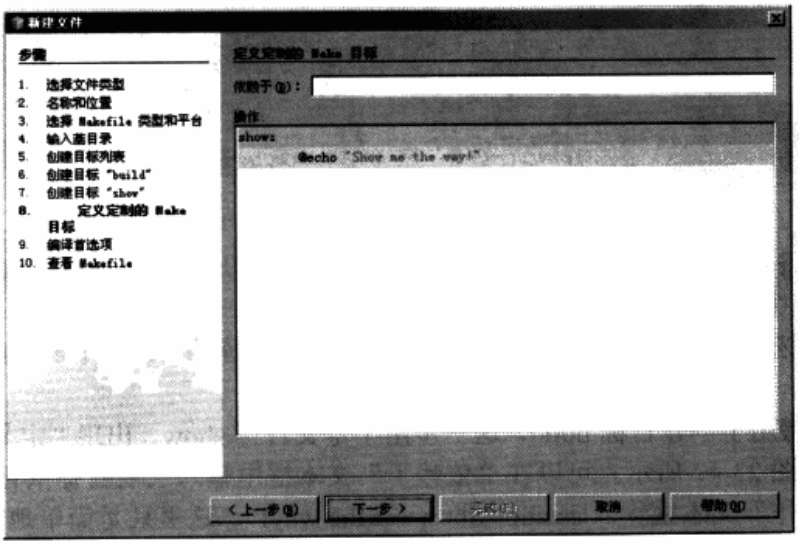


图 23-42 “定义定制的 Make 目标”对话框

Download at http://www.pin5i.com/

⑭ 单击“下一步”按钮，打开“设置基本选项”对话框，如图 23-43 所示。在此对话框中可以指定代码的生成方式。

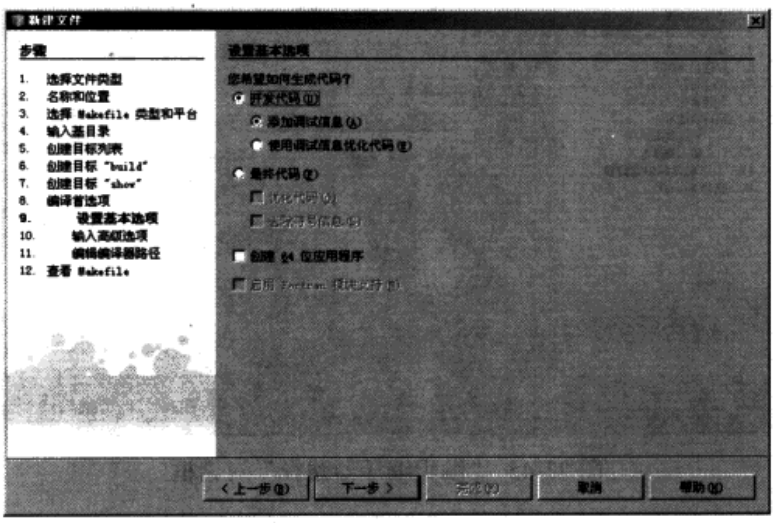


图 23-43 “设置基本选项”对话框

在选择“开发代码”的代码类型的情况，如果选择“添加调试信息”单选按钮，将产生一个包含调试信息的二进制文件；如果选择“使用调试信息优化代码”单选按钮，会使用调试信息来优化代码。若选择“最终代码”的代码类型的情况，可以根据需要选择“优化代码”或“去除符号信息”复选框。

如果要生成 64 位二进制文件，请选中“创建 64 位应用程序”复选框。如果选中“启用 Fortran 模块支持”复选框，则可以启用 Fortran 95 MODULE 语句支持。这里保持默认选项，单击“下一步”按钮。

⑮ 出现“输入高级选项”对话框，如图 23-44 所示。可以在该对话框中指定用于生成目标的编译器选项。各个高级选项字段均包含了一个 BASICOPTS 宏，该宏含有在“设置基本选项”页中选择的基本选项值。

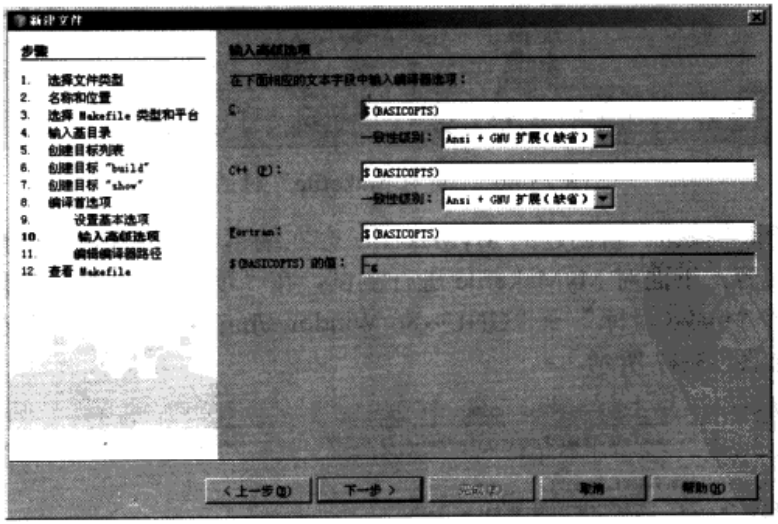


图 23-44 “输入高级选项”对话框

⑯ 这里不做修改，单击“下一步”按钮，打开“编辑编译器路径”对话框，如图 23-45 所示。在该对话框中可以指定用于生成目标的编译器路径。

Download at http://www.pin5i.com/

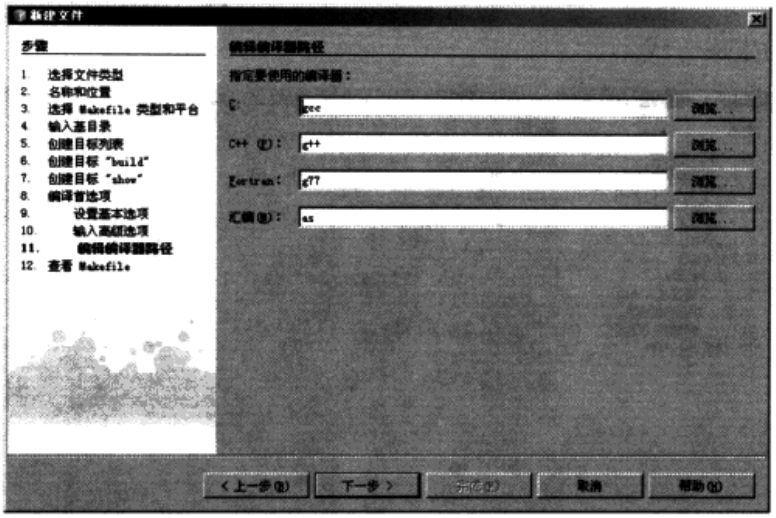


图 23-45 “编辑编译器路径”对话框

⑰ 这里保持默认选项，单击“下一步”按钮，打开“查看 Makefile”对话框，如图 23-46 所示。在该对话框中显示了 Makefile 的位置和名称、有关 Makefile 内容的详细信息及其使用说明。

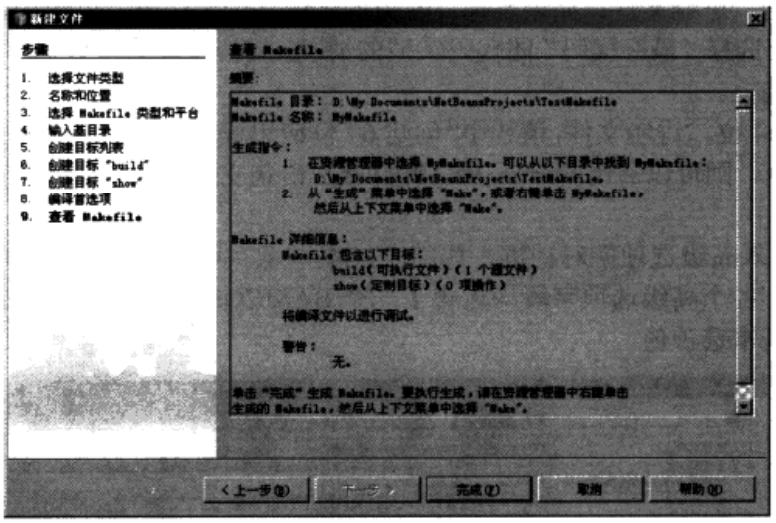


图 23-46 “查看 Makefile”对话框

⑱ 单击“完成”按钮。新创建的 MyMakefile 会在编辑器中打开，可以查看集成开发环境自动生成的代码内容。下面对 MyMakefile 进行测试，在“项目”窗口中，选择 MyMakefile 节点，右键单击选择“make 目标”→“GNU-x86-Windows/build”命令，可以在“输出”窗口中看到生成信息，如图 23-47 所示。



图 23-47 执行 build 目标的信息

Download at http://www.pin5i.com/

同理，在“项目”窗口中，选择 MyMakefile 节点，右键单击选择“make 目标”→“show”命令，可以在“输出”窗口中看到生成信息，如图 23-48 所示。

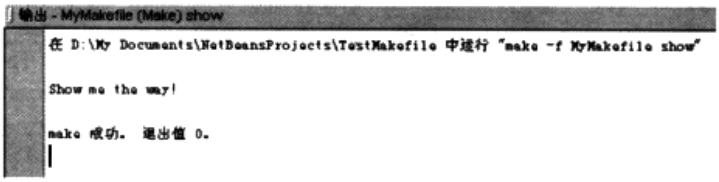


图 23-48 执行 show 目标的信息

23.6 创建基于现有代码的 C/C++项目

在 NetBeans 集成开发环境中，可以利用 C/C++ Pack 提供的向导创建基于现有代码的 C/C++项目。下面根据前面创建的 HelloWorld 项目创建新的 C/C++项目。

❶ 在 NetBeans 集成开发环境中，选择主菜单中的“文件”→“新建项目”命令，打开“新建项目”对话框。在“类别”列表中选择“C/C++”，在“项目”列表中选择“基于现有代码的 C/C++项目”。

❷ 单击“下一步”按钮，打开“生成工具”对话框，如图 23-49 所示。选择“使用现有的 Makefile”单选按钮，单击“现有的 Makefile”文本框后的“浏览”按钮，在打开的对话框中选择 HelloWorld 项目的 Makefile。

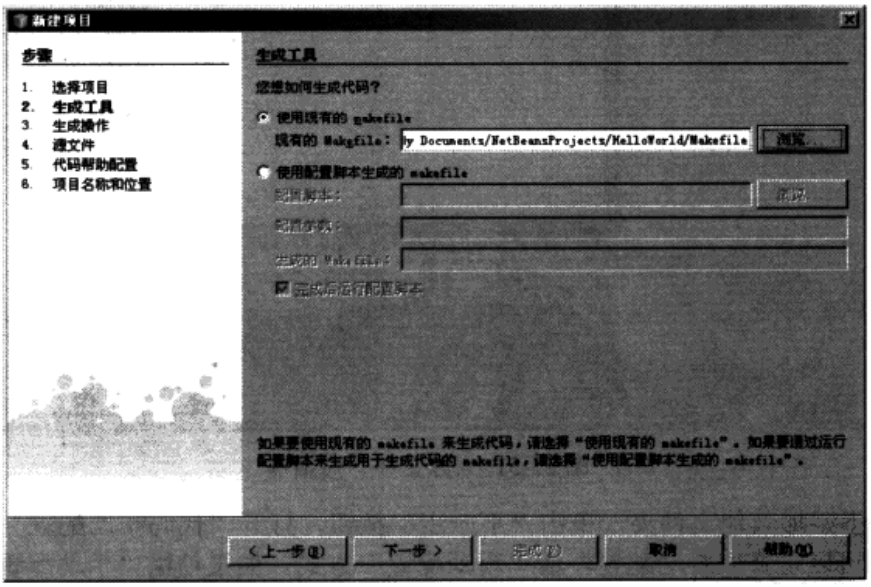


图 23-49 “生成工具”对话框

❸ 单击“下一步”按钮，打开“生成操作”对话框，如图 23-50 所示。“工作目录”、“生成命令”和“清理命令”文本框会被自动填充。在“生成结果”文本框中输入存放输出内容的完整路径，其结果可能是可执行文件也可能是库。该文本框中的内容可以保持为空等稍后再进行配置。

❹ 单击“下一步”按钮，打开“源文件”对话框，如图 23-51 所示。在此对话框中指定源项目哪些文件夹会被添加到新的项目中，向导会扫描项目文件夹中的源文件并将其添加到新的项目中。也可以通过单击“添加文件夹”按钮使用“选择源文件的文件夹”对话框来指定其他文件夹。还可以在“要添加到项目中的文件类型”下拉列表框中选择要添加的文件类型。

Download at http://www.pin5i.com/

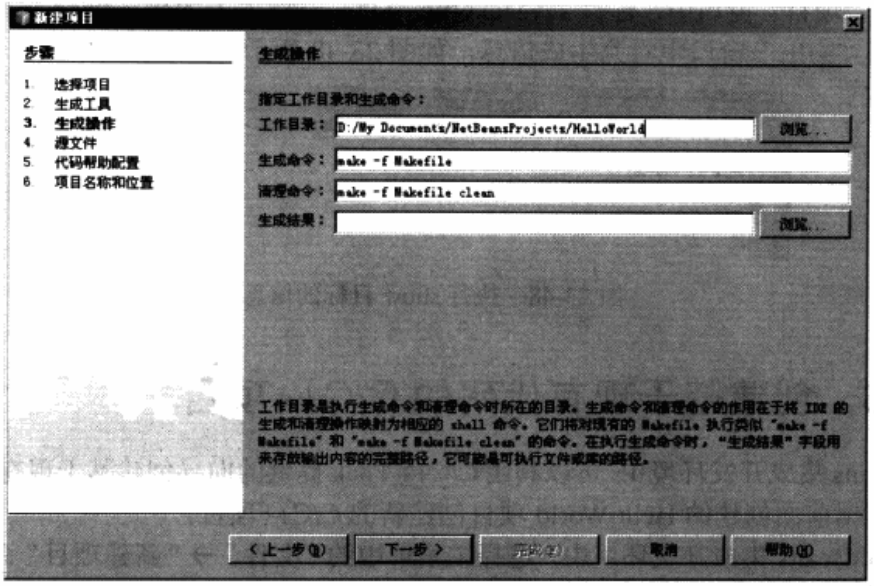


图 23-50 “生成操作”对话框

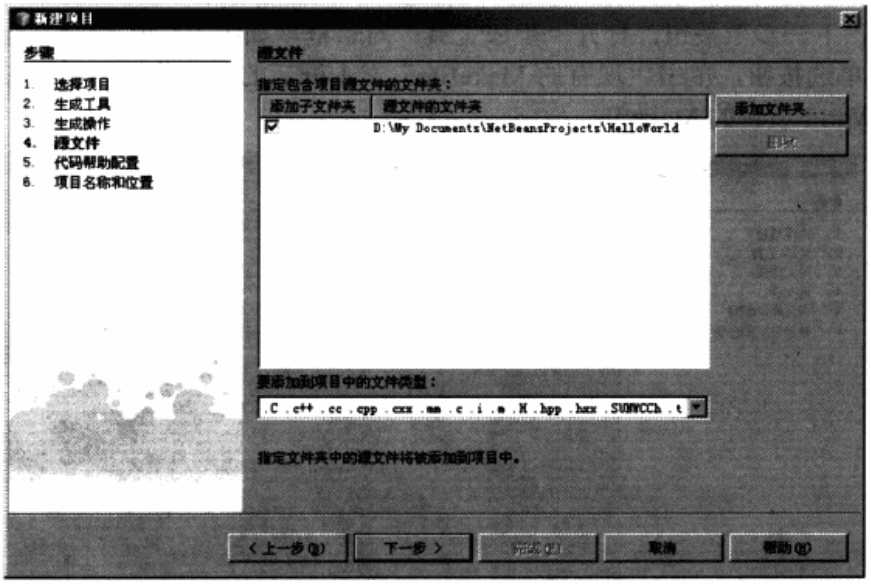


图 23-51 “源文件”对话框

❺ 这里不添加其他文件夹，单击“下一步”按钮，打开“代码帮助配置”对话框，如图 23-52 所示。可以选择“手动配置”或“自动配置”方式。如果使用“手动配置”方式，可以通过“编辑”按钮指定“包含目录”和“宏定义”，当解析源文件时，内置解析器将使用包含目录和宏定义。如果选择“自动配置”方式，需要指定其他的库，则会根据其他库自动发现项目配置。在“配置级别”下拉列表框中选择配置级别，可以是“配置项目属性”、“配置文件夹属性”或是“配置文件属性”。

❻ 这里不做任何修改，单击“下一步”按钮，打开“项目名称和位置”对话框，如图 23-53 所示。该对话框用于修改项目的名称，此处不做修改，直接单击“完成”按钮。

❼ 在“项目”窗口中，右键单击 HelloWorld_1 项目节点，选择弹出菜单中的“属性”命令。在打开的“项目属性”对话框中，选择“常规”节点，修改“配置类型”属性的值为“应用程序”。选择“正在运行”节点，修改“参数”属性的值为“cat dog pig bird”，单击“确定”按钮。

Download at http://www.pin5i.com/

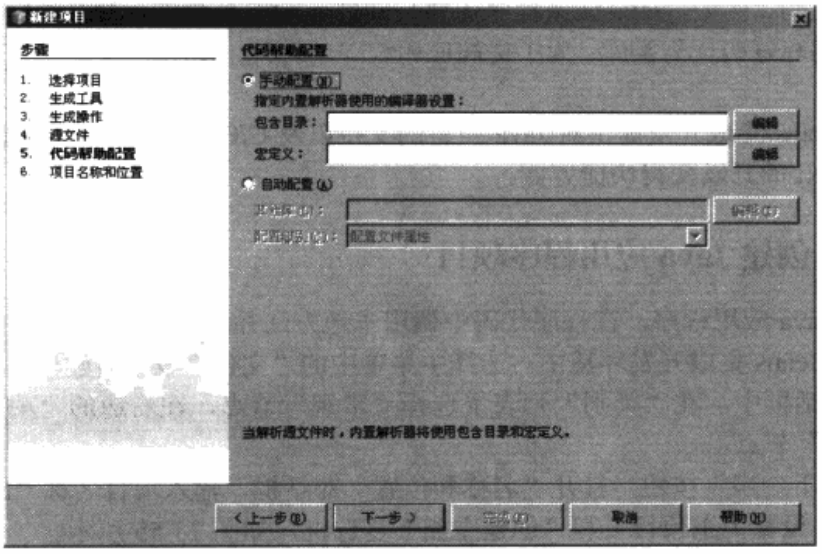


图 23-52 “代码帮助配置”对话框

❽ 单击工具栏上的“运行”按钮 ▶ 或按 F6 快捷键运行项目。运行结果如图 23-54 所示。

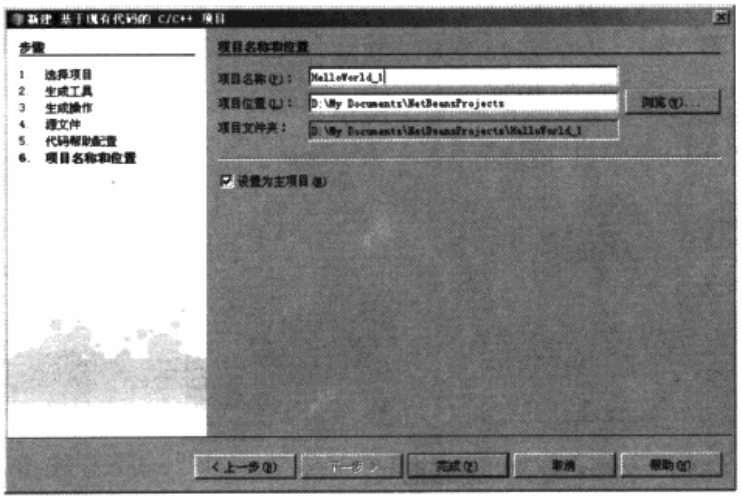


图 23-53 “项目名称和位置”对话框



图 23-54 项目运行结果

23.7 利用 C/C++ Pack 创建 JNI

通过将 Java 源代码编译成字节码并使用虚拟器解析，实现了跨平台的特性，但有时也对应用程序的性能有所限制，利用 JNI 实现了 Java 语言调用本地代码，从而实现和特定平台的交互。

23.7.1 JNI 简介

JNI 是 Java Native Interface 的缩写，即 Java 本地接口，是 Java 语言提供的一种通用接口，用于 Java 代码与其他像 C 和 C++这样的语言编写的本地化代码的交互。所谓本地化代码是指直接编译成的与机器相关的二进制代码，如 Windows 下面的可执行文件、DLL 等以及 Linux 下面的可执行文件和 SO 文件等。

Java 语言通过将代码编译成字节码这样的中间代码来实现代码的可移植性，但有些时候需

Download at http://www.pin5i.com/

要调用本地化代码。例如，Java 应用程序需要与硬件或操作系统进行交互；Java 应用程序使用其他语言已经编写好的优秀类库；为了提高应用程序的性能，使用汇编或 C 和 C++语言来实现部分功能。

利用 NetBeans 集成开发提供对 Java 语言的支持以及 C/C++ Pack 软件包提供对 C 和 C++ 语言的支持，JNI 的开发变得快捷方便。

23.7.2 创建 Java 应用程序项目

创建一个 Java 应用程序，在程序代码中调用本地方法并利用生成的类文件创建头文件。

❶ 在 NetBeans 集成开发环境中，选择主菜单中的“文件”→“新建项目”命令，打开的“选择项目”对话框中，在“类别”列表中选择“常规”节点，在右边的“项目”列表中选择“Java 应用程序”节点。

❷ 单击“下一步”按钮，打开“名称和位置”对话框，输入项目名称“Hello”并选择项目位置，选中“设置为主项目”和“创建主类”复选框，如图 23-55 所示。

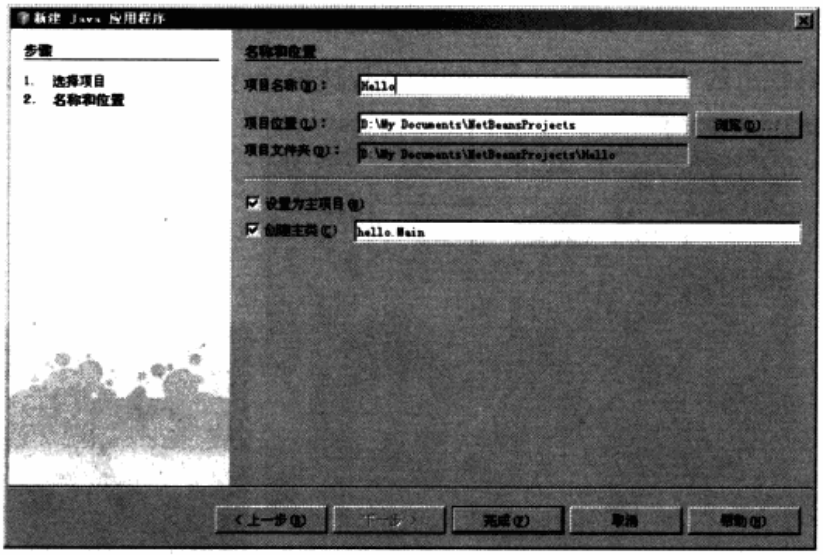


图 23-55 “名称和位置”对话框

❸ 单击“完成”按钮，自动创建 Main.java 文件。在源代码编辑器中打开该文件，并添加如下代码：

```
package hello;

/**
 *
 * @author admin
 */
public class Main {

    //此处静态游离块在加载类 Main 时用 System.loadLibrary()方法加载动态库
    static {
       System.loadLibrary("hello");//方法中的参数"hello"是生成的动态库的名字
    }

    private native void sayHello();//本地方法，用 native 关键字标识，没有方法体

    /** Creates a new instance of Main */
    public Main() {
    }
```

Download at http://www.pin5i.com/


```
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        new Main().sayHello();//创建 Main 对象实例并调用 sayHello()方法
    }
}
```

❹ 选择主菜单中的“生成”→“生成主项目”命令或按 F11 快捷键，生成项目。

❺ 打开 Windows 的命令提示符窗口，并转到项目根目录下的 build\classess 目录下，执行命令 javah -o hello.h hello.Main，之后执行 dir 命令会看到生成的 hello.h 头文件，如图 23-56 所示。

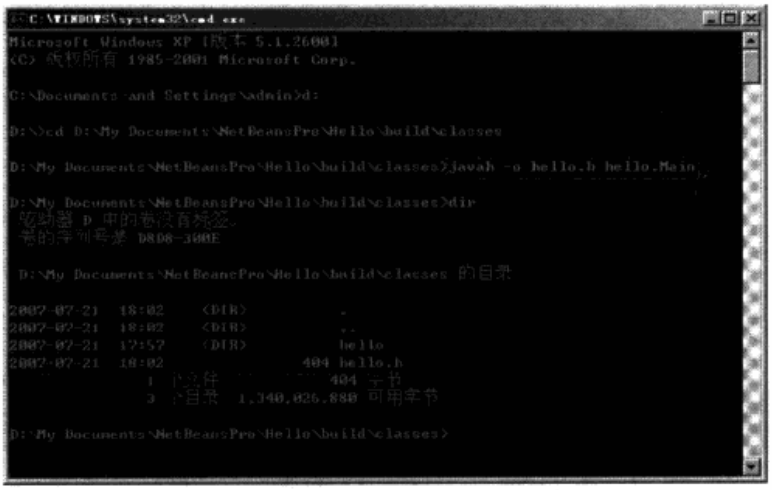


图 23-56 生成头文件

23.7.3 创建 C/C++动态库项目

创建一个 C 动态库项目，该项目代码实现在 Hello 项目中声明的 Java 本地方法 sayHello()。

❶ 在 NetBeans 集成开发环境中，选择主菜单中的“新建”→“新建项目”命令，打开的“选择项目”对话框。在“类别”列表中选择“C/C++”节点，在右边的“项目”列表中选择“C/C++动态库”节点，如图 23-57 所示。

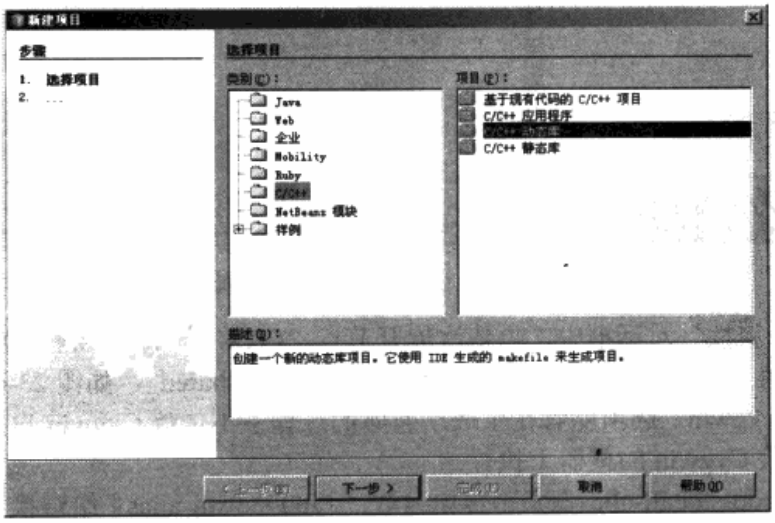


图 23-57 新建 C/C++动态库

Download at http://www.pin5i.com/

❷ 单击“下一步”按钮，打开“项目名称和位置”对话框。输入项目名称“HelloNative”并选择项目位置，其他保持默认设置，如图 23-58 所示。

❸ 单击“完成”按钮，创建 C/C++动态库项目。在“项目”窗口中，右键单击 HelloNative 项目节点，选择弹出菜单中的“属性”命令。在打开的“项目属性”对话框中，依次展开节点“C/C++”→“C 编译器”→“常规”。设置“包含目录”属性的值为<JAVA_HOME>/include;<JAVA_HOME>/include/win32，这是因为 C 源代码中需要 Java 库 jni.h。其中<JAVA_HOME>为 JDK 的安装路径，此路径不要有空格，否则会在生成过程产生错误。设置结果如图 23-59 所示。

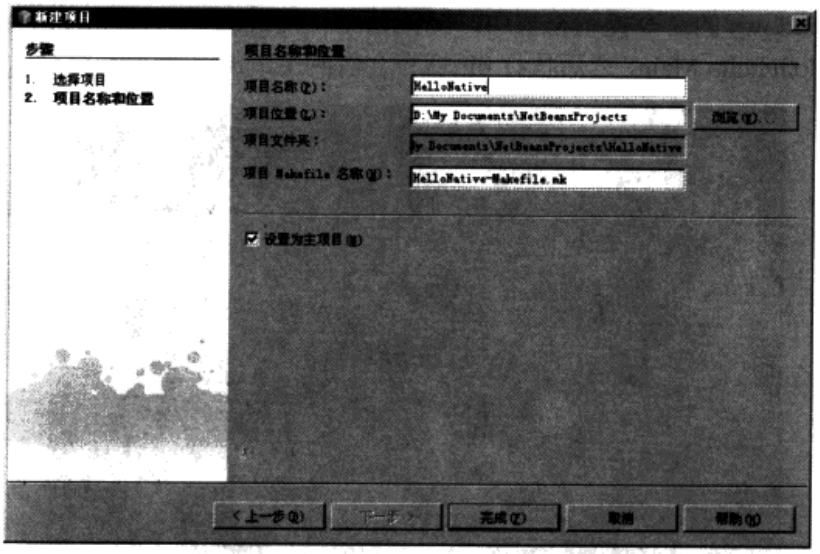


图 23-58 “项目名称和位置”对话框

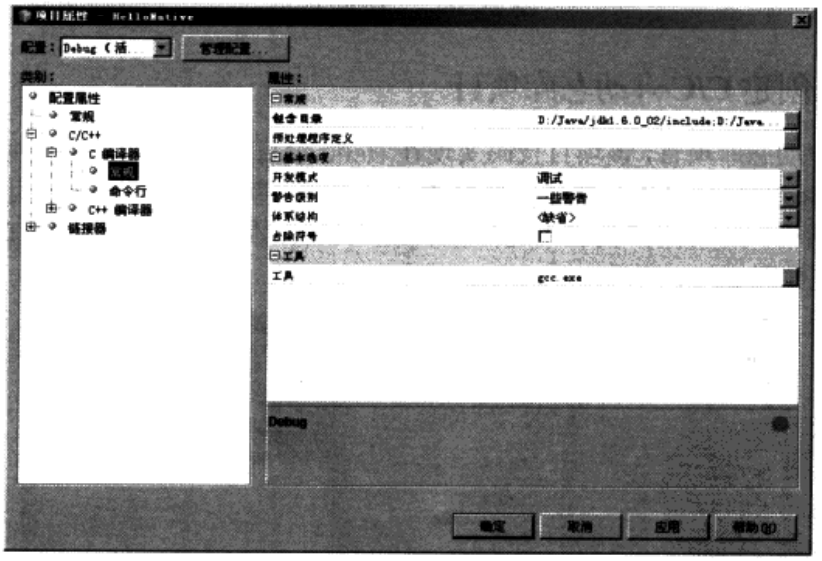


图 23-59 设置“包含目录”属性

❹ 在“项目属性”对话框中，再依次展开节点“C/C++”→“C 编译器”→“命令行”。设置其他选项的值为“-mno-cygwin -Wl,--add-stdcall-alias -shared”，如图 23-60 所示。

其中“-mno-cygwin”选项能够在生成动态库的过程中不依赖 Cygwin 自身的库，这样动态库能够在没有安装 Cygwin 的机器上执行。

选项“-Wl,--add-stdcall-alias”表示传递参数“--add-stdcall-alias”给链接器；如果没有这个选项，结果应用程序会产生 UnsatisfiedLinkError 错误而执行失败。

Download at http://www.pin5i.com/

“-shared”选项告知编译器生成一个动态链接库而不是一个可执行文件。

❺ 在“项目属性”对话框中，再依次展开节点“链接器”→“常规”。设置输出属性的值为“dist/Hello.dll”，这样指定生成的 DLL 文件的路径和名称，如图 23-61 所示。单击“确定”按钮完成项目属性的设置。

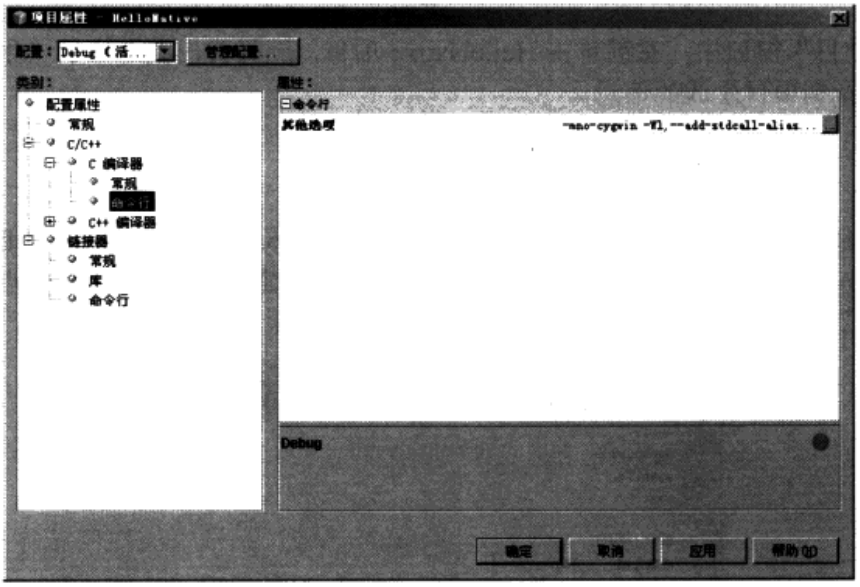


图 23-60 设置命令行选项

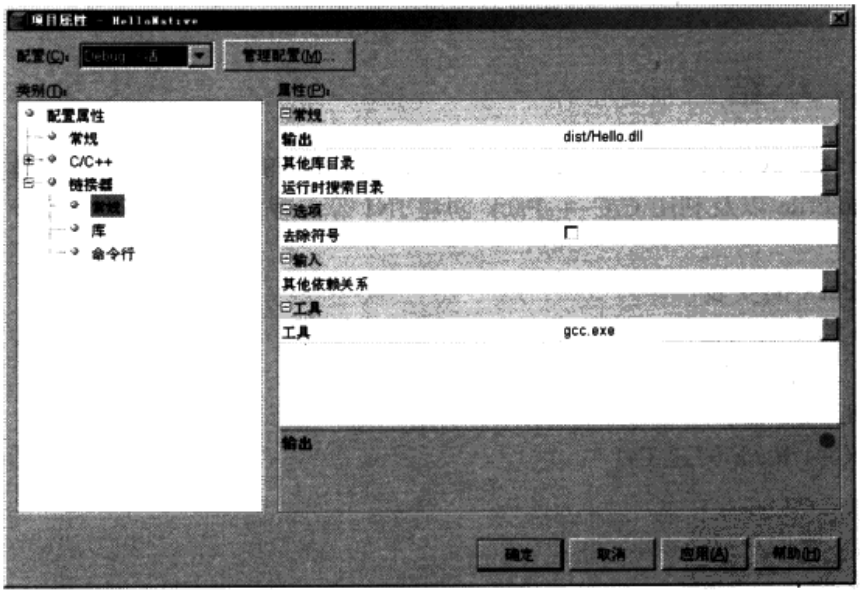


图 23-61 设置 DLL 文件输出

❻ 将前面生成的 hello.h 复制到 HelloNative 项目的根目录下。在 NetBeans 集成开发环境中，在“项目”窗口的 HelloNative 项目中的源文件节点上单击鼠标右键，选择弹出菜单中的“添加现有项”命令。在打开的“选择项”对话框中选择 HelloNative 项目根目录下的 hello.h。

❼ 在“项目窗口”中右键单击 HelloNative 项目节点下的源文件节点，选择弹出菜单中的“新建”→“空 C 文件”命令。在打开的“新建空 C 文件”对话框中，输入文件名称“HelloNative”，单击“完成”按钮。

❽ 编辑 HelloNative.c 文件，代码如下所示：

Download at http://www.pin5i.com/


```
#include <stdio.h>
#include "Hello.h"

JNIEXPORT void JNICALL Java_hello_Main_sayHello(JNIEnv *env, jobject obj){
    printf("\nHello JNI!\n");//打开输出字符串
    return;
}
```

❾ 在“项目”窗口中，右键单击 HelloNative 项目，选择弹出菜单中的“生成”命令，会在输出窗口中看到项目生成的信息。

23.7.4 生成并运行应用程序

将生成在 HelloNative 项目根目录下 dist 目录中的 Hello.dll 复制到 Hello 项目的根目录下。返回到 NetBeans 集成开发环境中，在“项目”窗口中，右键单击 Hello 项目节点，选择弹出菜单中的“运行”命令，可以在输出窗口中看到程序运行结果，如图 23-62 所示。

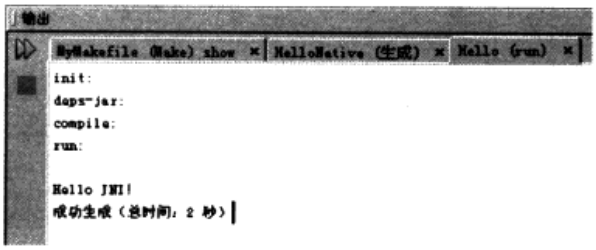


图 23-62 程序运行结果

23.8 小结

本章主要介绍C/C++ Pack的安装和设置、C和C++项目的创建及其属性的设置、设置C/C++ Pack、创建 Makefile 以及利用 C/C++ Pack 创建 JNI 等内容。通过本章的学习，读者应该掌握以下基本内容：

- C/C++ Pack 的安装
- Cygwin 的安装及设置
- C/C++项目的创建
- 设置 C/C++ Pack
- 利用 C/C++ Pack 创建 JNI

Download at http://www.pin5i.com/

附录 A

NetBeans 的快捷键



一、菜单快捷键

表 A-1 “文件”菜单

快捷键	命　令	描　述
Ctrl + Shift + N	新建项目	使用“新建项目”向导创建新项目
Ctrl + N	新建文件	使用“新建文件”向导创建新文件
Ctrl + Shift + O	打开项目	打开现有项目
Ctrl + O	打开文件	打开现有文件
Ctrl + S	保存	保存当前文件
Ctrl + P	打印	打印当前文件

表 A-2 “编辑”菜单

快捷键	命　令	描　述
Ctrl + Z	撤销	反向执行（一次一个）一系列编辑器操作，“保存”操作除外
Ctrl + Y	重做	反向执行（一次一个）一系列“撤销”命令
Ctrl + X	剪切	删除当前选择内容并将其放到剪贴板中
Ctrl + C	复制	将当前选择内容复制到剪贴板
Ctrl + V	粘贴	将剪贴板的内容粘贴到插入点处
Del	删除	删除当前选择内容
Ctrl + F	查找	查找文本字符串
Ctrl + H	替换	查找文本字符串并用指定的字符串进行替换
Alt + F7	查找使用实例	查找选定代码的使用实例和子类型
Ctrl + Shift + F	在项目中查找	在所有打开的项目中查找特定的时间段内修改的指定文本、对象名称和对象类型

表 A-3 “视图”菜单

快捷键	命　令	描　述
Ctrl + （数字键盘上的）-	折叠	编译项目的所有未编译和过期的类文件，即已更改或先前尚未编译的文件
Ctrl + （数字键盘上的）+	展开	编译当前项目的所有类及其依赖的类而不论它们先前是否被编译过
Ctrl + Shift + （数字键盘上的）-	全部折叠	指定项目的主类
Ctrl + Shift + （数字键盘上的）+	全部展开	运行当前项目
Alt + Shift + Enter	全屏	将窗口展开到屏幕的最大长度和宽度

Download at http://www.pin5i.com/

表 A-4 “导航”菜单

参见表 1-5 跳转命令。

表 A-5 “源”菜单

[illegible]	[illegible]	[illegible]
Alt + Shift + F	格式	设置选定代码或整个文件（如果未选择任何内容）的格式
Alt + Shift + 向左方向键	左移	将选定的一行或几行向左移动一个制表位
Alt + Shift + 向右方向键	右移	将选定的一行或几行向右移动一个制表位
Alt + Shift + 向上方向键	上移	将选定的一行或几行上移一行
Alt + Shift + 向下方向键	下移	将选定的一行或几行下移一行
Ctrl + Shift + Up	向上复制	将选定的一行或几行复制到上一行
Ctrl + Shift + Down	向下复制	将选定的一行或几行复制到下一行
Ctrl + 斜杠	切换注释	打开或关闭当前行或选定行的注释
Ctrl + 反斜杠	完成代码	显示代码完成框
Alt + Insert	插入代码	弹出一个上下文相关菜单，可以使用此菜单来生成通用结构，如构造函数、getter 和 setter
Alt + Enter	修复代码	显示编辑器提示。如果在显示灯泡时有相应提示，IDE 将通知进行修复
Ctrl + Shift + I	修复导入	生成文件中指定的类所需的 import 语句
Ctrl + P	显示方法参数	选择下一个参数。要使此快捷键起作用，必须选中（突出显示）一个参数
Ctrl + Shift + 空格键	显示文档	显示光标所在位置的项的文档
Ctrl + Shift + K	插入下一个匹配词语	键入代码的开头字符时，将在代码的其他位置生成所使用的下一个词
Ctrl + K	插入上一个匹配词语	键入代码的开头字符时，将在代码的其他位置生成所使用的上一个词

表 A-6 “生成”菜单

[illegible]	[illegible]	[illegible]
F11	生成主项目	编译文件。如果选择文件夹，则 IDE 会编译所有文件而无论自最近一次编译以来它们是否发生更改
Shift + F11	清理并生成主项目	编译文件。如果选择文件夹，则 IDE 会编译所有文件而无论自最近一次编译以来它们是否发生更改
F9	编译文件	编译文件。如果选择文件夹，则 IDE 只编译新的或自最近一次编译以来发生更改的文件

表 A-7 “运行”菜单

[illegible]	[illegible]	[illegible]
F6	运行主项目	运行主项目
Ctrl + F5	调试主项目	调试主项目
Alt + F6	测试项目	启动项目的 JUnit 测试
Shift + F6	运行文件	启动项目的 JUnit 测试
Ctrl + Shift + F5	调试文件	启动当前文件的调试会话

Download at http://www.pin5i.com/

续表

Ctrl + F6	测试文件	启动当前文件的 JUnit 测试
Ctrl + Shift + F6	调试文件的测试	在 JUnit 中启动文件的调试测试
Shift + F5	结束调试器会话	结束调试会话
F5	继续	恢复调试直至到达下一断点或程序末尾
F8	越过	执行一行程序源代码。如果此行是方法调用，则执行整个方法，然后停止
Shift + F8	越过表达式	越过表达式，然后停止调试
F7	步入	执行一行程序源代码。如果此行是方法调用，则执行程序直至到达方法的第一条语句，然后停止
Ctrl + F7	步出	执行一行程序源代码。如果此行是方法调用，则执行方法，然后将控制权返回给调用者
F4	运行至光标	将当前项目运行到文件中的光标位置，然后停止程序执行
Ctrl + Alt + 向上方向键	激活被调用者	使被调用的方法成为当前调用。此命令仅当在“调用栈”窗口中选定调用时才可用
Ctrl + Alt + 向下方向键	激活调用者	使调用方法成为当前调用。此命令仅当在“调用栈”窗口中选定调用时才可用
Ctrl + F8	切换断点	添加一个行断点，或删除程序中光标位置处的断点
Ctrl + Shift + F8	新建断点	在指定行、异常或方法中设置新断点
Ctrl + Shift + F7	新建监视	添加指定的监视变量
Ctrl + F9	计算表达式的值	打开“计算表达式的值”对话框

表 A-8 “工具”菜单

Ctrl + Shift + J	插入国际化字符串	允许在创建源代码时添加国际化字符串

表 A-9 “窗口”菜单

Ctrl + 0	源代码编辑器	切换到源代码编辑器
Ctrl + 1/Ctrl + Shift + 1	项目	打开“项目”窗口
Ctrl + 2/Ctrl + Shift + 2	文件	打开“文件”窗口
Ctrl + 3/Ctrl + Shift + 3	收藏夹	打开“收藏夹”窗口
Ctrl + 4	“输出”窗口	打开“输出”窗口
Ctrl + 5	运行环境	打开“运行环境”窗口
Ctrl + Shift + 5	HTTP 监视器	打开 HTTP 监视器
Ctrl + 6	任务列表	打开“任务列表”窗口
Ctrl + 7	导航	打开“导航”窗口
Alt + Shift + 1	“调试”>“局部变量”	打开“局部变量”调试器窗口
Alt + Shift + 2	“调试”>“监视”	打开“监视”调试器窗口
Alt + Shift + 3	“调试”>“调用栈”	打开“调用栈”调试器窗口
Alt + Shift + 4	“调试”>“类”	打开“类”调试器窗口
Alt + Shift + 5	“调试”>“断点”	打开“断点”调试器窗口

Download at http://www.pin5i.com/

续表

快捷键	菜单	描述
Alt + Shift + 6	“调试”>“会话”	打开“会话”调试器窗口
Alt + Shift + 7	“调试”>“线程”	打开“线程”调试器窗口
Alt + Shift + 8	“调试”>“源”	打开“源”窗口
Ctrl + Tab (Ctrl + ')	切换到最近使用的窗口	按最近使用顺序在打开的窗口之间进行切换。对话框显示源代码编辑器中所有打开的窗口和每个打开的文档
Shift + Esc	最大化窗口	最大化源代码编辑器或当前窗口
Ctrl + W	关闭	关闭当前窗口中的当前标签。如果窗口没有标签，则关闭整个窗口
Ctrl + Shift + F4	关闭所有文档	关闭源代码编辑器中所有打开的文档
Shift + F4	文档	打开“文档”对话框，可以在该对话框中保存和关闭打开的文档组
Alt + 向右方向键	下一个标签	显示当前窗口中的下一个标签
Alt + 向左方向键	上一个标签	显示当前窗口中的上一个标签

表 A-10 “帮助”菜单

快捷键	菜单	描述
F1	帮助	显示 JavaHelp 查看器，其中含有所有已安装的帮助集
Shift + F1	Javadoc 索引搜索	搜索 Javadoc 索引

二、编辑器快捷键

表 A-11 滚动和选择

快捷键	描述
Ctrl + 向下箭头键	向上滚动窗口但不移动插入点
Ctrl + 向上箭头键	向下滚动窗口但不移动插入点
Ctrl + [	将插入点移到与突出显示的方括号匹配的方括号后。仅当插入点紧跟在左方括号或右方括号之后时，此快捷键才起作用
Ctrl + Shift + [	选择一对方括号之间的块。仅当插入点紧跟在左方括号或右方括号之后时，此快捷键才起作用
Ctrl + G	跳转至任何指定的行
Ctrl + Home	将插入点移到文件开始处
Ctrl + End	将插入点移到文件结尾处
Ctrl + A	选择文件中所有的文本

表 A-12 修改文本

快捷键	描述
Insert	在插入文本和覆盖文本模式之间切换
Ctrl + Shift + J	打开“国际化”对话框，可以使用该对话框在插入点插入国际化字符串
Ctrl + U, U	使选定字符或插入点右边字符变为大写字母
Ctrl + U, L	使选定字符或插入点右边字符变为小写字母
Ctrl + U, S	颠倒选定字符或插入点右边字符的大小写

Download at http://www.pin5i.com/

表 A-13 代码折叠

[illegible]	[illegible]
Ctrl + 减号 (-)	折叠插入点所在的代码块
Ctrl + 加号 (+)	展开插入点旁边的代码块
Ctrl + Shift + 减号 (-)	折叠所有代码块
Ctrl + Shift + 加号 (+)	展开所有代码块

表 A-14 复制、粘贴和删除文本

[illegible]	[illegible]
Ctrl + Z	（撤销）反向执行（一次一个）一系列的编辑器操作，“保存”除外
Ctrl + Y	（重做）反向执行（一次一个）一系列的“撤销”命令
Ctrl + X	（剪切）删除当前选择并将其放到剪贴板中
Shift + Delete	（剪切）删除当前选择并将其放到剪贴板中
Ctrl + C	（复制）将当前选择复制到剪贴板
Ctrl + Insert	（复制）将当前选择复制到剪贴板
Ctrl + V	（粘贴）将剪贴板的内容粘贴到插入点处
删除	（删除）删除当前选择
Ctrl + E	删除当前行
Ctrl + Backspace	删除当前词语中插入点之前的文本

表 A-15 搜索文本

[illegible]	[illegible]
Ctrl + F3	搜索插入点所在处的词语并突出显示找到的所有这个词语
F3	选择当前搜索词语的下一个出现位置
Shift + F3	选择当前搜索词语的上一个出现位置
Alt + Shift + H	打开或关闭搜索结果的突出显示
Ctrl + F	打开“查找”对话框
Ctrl + H	打开“查找和替换”对话框

表 A-16 设置制表符

[illegible]	[illegible]
Tab	向右移动插入点右边的所有文本
Alt + Shift + 向右方向键	向右移动在包含插入点的行中的文本
Alt + Shift + 向左方向键	向左移动在包含插入点的行中的文本

三、窗口导航快捷键

表 A-17 窗口导航快捷键

[illegible]	[illegible]
Tab	将焦点转移到下一个组件或下一组组件中的第一个成员（按从左到右的顺序，左上方的组件）
Shift + Tab	将焦点转移到上一个组件或上一组组件中的最后一个成员（Tab 的反向浏览）

Download at http://www.pin5i.com/

续表

快捷键	描述
Home	选择列表或窗格中的第一项
End	选择列表或窗格中的最后一项
向上箭头键	将键盘焦点移到项目组中的前一项。导航至属性表单组合框中的前一个设置
向下箭头键	将键盘焦点移到项目组中的后一项。导航至属性表单组合框中的后一个设置
向左箭头键	关闭文件夹（节点）
向右箭头键	打开文件夹（节点）
空格键	激活具有焦点的项目
Ctrl + 0	切换到源代码编辑器
Ctrl + 1/Ctrl + Shift + 1	打开“项目”窗口
Ctrl + 2/Ctrl + Shift + 2	打开“文件”窗口
Ctrl + 3/Ctrl + Shift + 3	打开“收藏夹”窗口
Ctrl + 4	打开“输出”窗口
Ctrl + 5	打开“运行环境”窗口
Ctrl + 6	打开“任务列表”窗口
Ctrl + 7	打开 HTTP 监视器
Alt + Shift + 1	打开“局部变量”调试器窗口
Alt + Shift + 2	打开“监视”调试器窗口
Alt + Shift + 3	打开“调用栈”调试器窗口
Alt + Shift + 4	打开“类”调试器窗口
Alt + Shift + 5	打开“断点”调试器窗口
Alt + Shift + 6	打开“会话”调试器窗口
Alt + Shift + 7	打开“线程”调试器窗口
Alt + Shift + 8	打开“源”窗口
Ctrl + Tab (Ctrl + ')	按最近使用顺序在打开的窗口之间进行切换。对话框显示源代码编辑器中所有打开的窗口和每个打开的文档
Shift + Esc	最大化源代码编辑器或当前窗口
Ctrl + W	关闭当前窗口中的当前标签。如果窗口没有标签，则关闭整个窗口
Ctrl + Shift + F4	关闭源代码编辑器中所有打开的文档
Shift + F4	打开“文档”对话框，可以在该对话框中保存和关闭打开的文档组
Alt + 向右箭头键	显示当前窗口中的下一个标签
Alt + 向左箭头键	显示当前窗口中的上一个标签

四、帮助查看器快捷键

表 A-21 帮助查看器快捷键

快捷键	描述
Ctrl + Tab	将焦点移到下一个组件
Ctrl + Shift + Tab	将焦点移到上一个组件
向上箭头键	选择目录或索引中的上一项

Download at http://www.pin5i.com/

续表

快捷键	描　述
向上箭头键	移动分路器栏
向下箭头键	选择目录或索引中的下一项
	移动分路器栏
向左箭头键	将焦点转移到上一个制表符或上一个按钮
	关闭目录或索引中的文件夹
	向左移动分路器栏
向右箭头键	将焦点转移到下一个制表符或下一个按钮
	打开目录或索引中的文件夹
	向右移动分路器栏
Ctrl + 向左箭头	向左滚动
Ctrl + 向右箭头	向右滚动
Home	选择目录或索引中的第一项
End	选择目录或索引中的最后一项
Page Up	向上滚动一页
Page Down	向下滚动一页
Ctrl + T	将焦点移到主题的下一个链接
Ctrl + Shift + T	将焦点移到主题的上一个链接
Ctrl + 空格键	跟踪主题中的链接
空格键	激活具有焦点的工具栏按钮（“上一页”、“下一页”、“打印”或“页面设置”）
Ctrl + F1	显示工具栏按钮的交替文本
F6	在左右窗格之间转移焦点
F8	如果焦点在内容窗格或在浏览窗格中，请选择分路器栏

五、设置快捷键

在 NetBeans 中快捷键还可以根据自己的习惯进行定制。在主窗口中选择“工具”→“选项”命令，然后单击“快捷键映射”按钮，打开定制快捷键的界面，如图 A-1 所示。

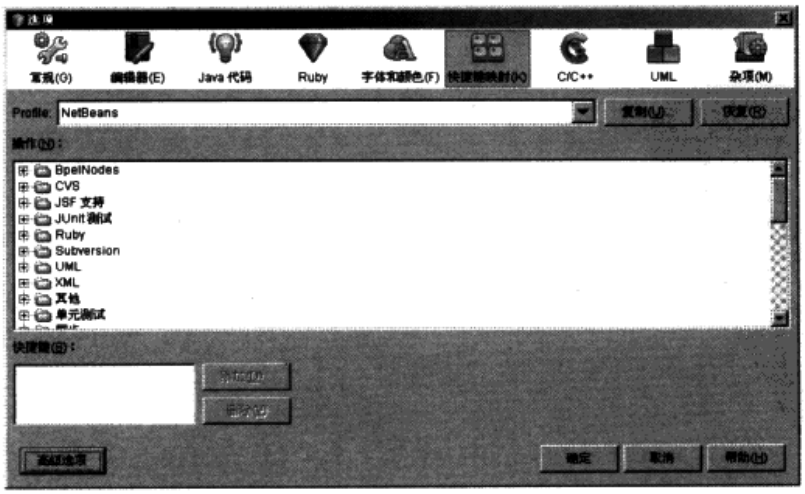


图 A-1　定制快捷键

Download at http://www.pin5i.com/

“Profile”下拉列表框中可以选择快捷键的配置文件，可以使用 Eclipse、Emacs 和 NetBeans 的快捷键配置文件。在“操作”列表中选中要更改的命令，然后单击“删除”按钮删除当前的快捷键。单击“添加”按钮，打开“添加快捷键”对话框，如图 A-2 所示。

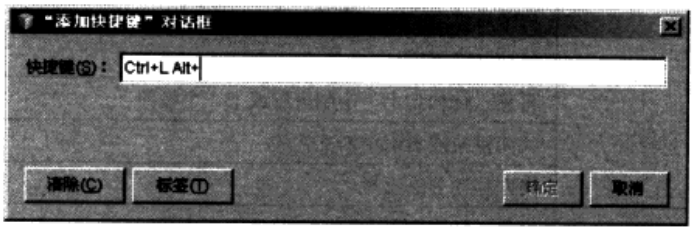


图 A-2 “添加快捷键”对话框

在“快捷键”文本框中按下任意的按键组合，会自动地记录并显示在“快捷键”文本框中。单击“确定”按钮，就会为指定的命令添加一个快捷键。